User's Manual

32

The revision list summarizes the locations of revisions and additions. Details should always be checked by referring to the relevant text.

SH7256 Group

User's Manual: Hardware

Renesas 32-Bit RISC Microcomputer
SuperH™ RISC engine Family

SH72567R R5F72567RKBGV

Renesas Electronics
www.renesas.com

Rev.3.00 Apr 2014

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
3. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.

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

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.

   Renesas Electronics products are neither intended nor authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems, surgical implantations etc.), or may cause serious property damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
11. This document may not be reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

(2012.4)

General Precautions on Handling of Product

1. Treatment of NC Pins

Note: Do not connect anything to the NC pins.
The NC (not connected) pins are either not connected to any of the internal circuitry or are used as test pins or to reduce noise. If something is connected to the NC pins, the operation of the LSI is not guaranteed.

2. Treatment of Unused Input Pins

Note: Fix all unused input pins to high or low level.
Generally, the input pins of CMOS products are high-impedance input pins. If unused pins are in their open states, intermediate levels are induced by noise in the vicinity, a pass-through current flows internally, and a malfunction may occur.

3. Processing before Initialization

Note: When power is first supplied, the product's state is undefined.
The states of internal circuits are undefined until full power is supplied throughout the chip and a low level is input on the reset pin. During the period where the states are undefined, the register settings and the output state of each pin are also undefined. Design your system so that it does not malfunction because of processing while it is in this undefined state. For those products which have a reset function, reset the LSI immediately after the power supply has been turned on.

4. Prohibition of Access to Undefined or Reserved Addresses

Note: Access to undefined or reserved addresses is prohibited.
The undefined or reserved addresses may be used to expand functions, or test registers may be allocated to these addresses. Do not access these registers; the system's operation is not guaranteed if they are accessed.

5. Reading from/Writing to Reserved Bit of Each Register

Note: Treat the reserved bit of register used in each module as follows except in cases where the specifications for values which are read from or written to the bit are provided in the description.
The bit is always read as 0. The write value should be 0 or one, which has been read immediately before writing.
Writing the value, which has been read immediately before writing has the advantage of preventing the bit from being affected on its extended function when the function is assigned.

Configuration of This Manual

This manual comprises the following items:

1. General Precautions on Handling of Product
2. Configuration of This Manual
3. Preface
4. Contents
5. Overview
6. Description of Functional Modules
   - CPU and System-Control Modules
   - On-Chip Peripheral Modules

   The configuration of the functional description of each module differs according to the module. However, the generic style includes the following items:

   i) Feature
   ii) Input/Output Pin
   iii) Register Description
   iv) Operation
   v) Usage Note

When designing an application system that includes this LSI, take notes into account. Each section includes notes in relation to the descriptions given, and usage notes are given, as required, as the final part of each section.

7. List of Registers
8. Electrical Characteristics
9. Appendix
10. Main Revisions and Additions in this Edition (only for revised versions)

The list of revisions is a summary of points that have been revised or added to earlier versions. This does not include all of the revised contents. For details, see the actual locations in this manual.

11. Index

Preface

This LSI is an RISC (Reduced Instruction Set Computer) microcontroller that includes a Renesas original RISC CPU as its core, and the peripheral functions required to configure a system.

Target Users: This manual was written for users who will be using this LSI in the design of application systems. Target users are expected to understand the fundamentals of electrical circuits, logical circuits, and microcontrollers.

Objective: This manual was written to explain the hardware functions and electrical characteristics of this LSI to the target users.
Refer to the SH-2A, SH2A-FPU Software Manual for a detailed description of the instruction set.

Notes on reading this manual:

- In order to understand the overall functions of the chip

  Read the manual according to the contents. This manual can be roughly categorized into parts on the CPU, system control functions, peripheral functions and electrical characteristics.
- In order to understand the details of the CPU's functions

  Read the SH-2A, SH2A-FPU Software Manual.
- In order to understand the details of a register when its name is known

  Read the index that is the final part of the manual to find the page number of the entry on the register. The addresses, bits, and initial values of the registers are summarized in section 33, List of Registers.

- Examples

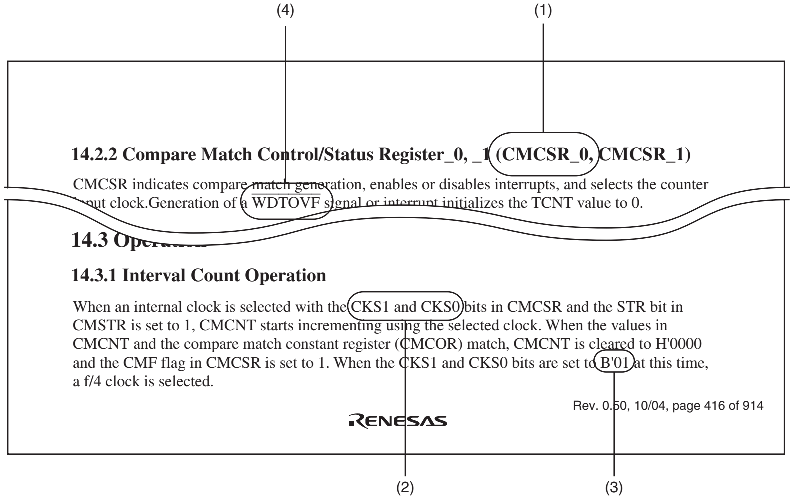
  The notation used for register names, bit names, numbers, and symbols in this manual is described below.

(1) Registers
The style (register name)_(channel number) is used in cases where the same or a similar function is implemented on more than one channel.
Example: CMCSR_0

(2) Bits
When bit names are given in this manual, the higher-order bits are to the left and the lower-order bits are to the right.
Example: CKS1, CKS0

(3) Numbers
Binary numbers are given as B'xxxx, hexadecimal are given as H'xxxx, and decimal are given as xxxx.
Examples: B'11 or 11, H'EFA0, 1234

(4) Symbols
An overbar is added to the names of active-low signals.
Example: $\overline{\text{WDTOVF}}$



14.2.2 Compare Match Control/Status Register_0, _1 (CMCSR_0, CMCSR_1)

CMCSR indicates compare match generation, enables or disables interrupts, and selects the counter input clock.Generation of a $\overline{\text{WDTOVF}}$ signal or interrupt initializes the TCNT value to 0.

14.3 Operation

14.3.1 Interval Count Operation

When an internal clock is selected with the CKS1 and CKS0 bits in CMCSR and the STR bit in CMSTR is set to 1, CMCNT starts incrementing using the selected clock. When the values in CMCNT and the compare match constant register (CMCOR) match, CMCNT is cleared to H'0000 and the CMF flag in CMCSR is set to 1. When the CKS1 and CKS0 bits are set to B'01 at this time, a f/4 clock is selected.

Rev. 0.50, 10/04, page 416 of 914

RENESAS

Note: The bit names and sentences in the above figure are examples, and have nothing to do with the contents of this manual.

- Notation in bit figures and tables describing arrangements of bits

  Each register description includes a figure that illustrates the arrangement of bits and a table that describes the meanings of settings in the bits.

(1) Bit

Indicates the bit number.

In the case of a 32-bit register, the bits are arranged in order from 31 to 0, and in the case of a 16-bit register, the bits are arranged in the order from 15 to 0.

(2) Bit Name

The short form of the name of the bit or bit field within the register.

When the individual bits of bit fields have to be clearly indicated, notation allowing this is included (e.g., ASID[3:0]).

A reserved bit is indicated by –.

Instead of a bit name, a blank is used for some bits, such as those of timer counters.

(3) Initial Value

Indicates the value of each bit after a power-on reset, i.e., the initial value.

0: Initial value is 0

1: Initial value is 1

–: Initial value is undefined

(4) R/W

Indicates whether each bit is readable or writable, or either writing to or reading from the bit is prohibited.

The notation is as follows:

R/W: Bit or field is readable and writable.

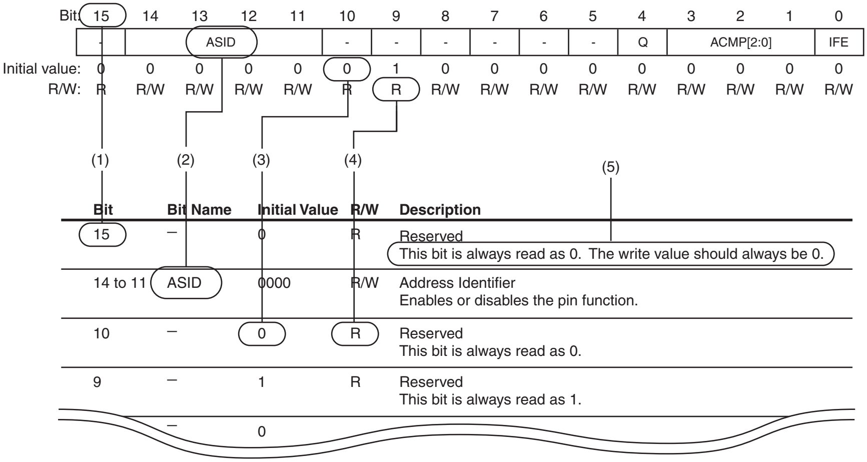
R/(W): Bit or field is readable and writable.
However, writing is only performed to clear the flag.

R: Bit or field is readable and writable.
However, "R" is indicated for all reserved bits. When writing to the bit is required, write the value stated in the bit table or the initial value.

W: Bit or field is readable and writable.
However, only the value in the bit table is guaranteed when reading from the bit.

(5) Description

Describes the function enabled by setting the bit.



Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	ASID				-	-	-	-	-	-	Q	ACMP[2:0]			IFE
Initial value:	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	–	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 11	ASID	0000	R/W	Address Identifier Enables or disables the pin function.
10	–	0	R	Reserved This bit is always read as 0.
9	–	1	R	Reserved This bit is always read as 1.
	–	0		

Note: The bit names and sentences in the above figure are examples, and have nothing to do with the contents of this manual.

All trademarks and registered trademarks are the property of their respective owners.

Contents

Section 26 ROM 1583

Section 27 Intelligent Flash Security (IFS) 1703

Section 28 EEPROM 1727

Section 1 Overview

1.1 Features of the SH7256 Group

The SH7256 group is a single-chip microcontroller featuring an original Renesas RISC (reduced instruction set computer) SH-2A-type CPU core along with integrated peripheral functions essential to system configuration.

This CPU brings the user the ability to set up high-performance systems with strong functionality at less expense than was achievable with previous microcontrollers, and is even able to handle real-time control applications requiring high-speed characteristics. Superscalar architecture and a Harvard cache architecture further improve the CPU's characteristics and performance.

The SH-2A has a 32-bit RISC architecture, and features upward compatibility with the SH-2E core at the object code level. Furthermore, the new instructions added to the instruction set for the existing SH-2E core contribute to higher speeds of execution and greater code efficiency.

On-chip peripheral functions essential to system configuration on this LSI include a floating point unit (FPU), large-capacity ROM and RAM, a direct memory access controller (DMAC), timers, an automotive direct memory access controller (A-DMAC), a Renesas serial peripheral interface (RSPI), an user break controller (UBC), an advanced user debugger II (AUD-II), an advanced timer unit III (ATU-III), a serial communications interface (SCI), a controller area network interface (RCAN-TL1), a FlexRay module, an A/D converter (ADC), an interrupt controller (INTC), and I/O ports.

This LSI also includes functions for external access control that allow the direct connection of various external memory modules and peripheral devices, and has the potential to greatly reduce system costs.

The on-chip ROM of this LSI takes the form of flexible zero turn-around time (F-ZTAT) memory, i.e. flash memory. Flash memory can be reprogrammed and erased by a user programmer or other software. This allows the user to reprogram the chip while it is mounted on the board.

The features of this LSI are listed in table 1.1.

Note: F-ZTAT is a trademark of Renesas Electronics Corporation.

Table 1.1 Features of the SH7256 Group

Item	Features
CPU	• Original Renesas SuperH architecture • Upwardly compatible with SH-1, SH-2, and SH-2E cores on the object code level • 32-bit internal data bus • General-register architecture Sixteen 32-bit general registers Four 32-bit control registers Four 32-bit system registers Register banks for fast interrupt response • RISC-type instruction set (upward-compatible with SH-2E Series) Instruction length: 16-bit basic instructions for improved code efficiency, and 32-bit instructions for improved performance and ease of use Load-store architecture Delayed branch instructions Instruction set based on C language • Superscalar architecture allowing simultaneous execution of two instructions, including FPU • Instruction execution time: Max. 2 instructions/cycle • Address space: 4 Gbytes • On-chip multiplier • 5-stage pipeline • Harvard architecture

Item	Features
Floating-point unit (FPU)	• On-chip floating-point coprocessor • Supports single-precision (32 bits) and double-precision (64 bits) • Supports IEEE 754-compliant data types and exceptions • 2 rounding modes: Round to Nearest and Round to Zero • Handling of denormalize numbers: Truncation to zero • Floating-point registers Sixteen 32-bit floating-point registers (single-precision × 16 words or double-precision × 8 words) Two 32-bit floating-point system registers • Supports FMAC (multiply and accumulate) instruction • Supports FDIV (division) and FSQRT (square root) instructions • Supports FLDI0/FLDI1 (load constant 0/1) instructions • Instruction execution times Latency (FMAC/FADD/FSUB/FMUL): 3 cycles (single-precision), 8cycles (double-precision) Pitch (FMAC/FADD/FSUB/FMUL): 1 cycle (single-precision), 6cycles (double-precision) Note: FMAC is supported for single-precision only. • 5-stage pipeline • Selectable exception generation when qNaN or ±∞ input

Item	Features
Operating modes	• Operating modes Single-chip mode 8/16-bit bus extended mode — On-chip ROM enabled — On-chip ROM disabled • On-board programming mode Boot mode User boot mode User program mode • Processing states Reset state Program execution state Exception handling state • Power-down modes Sleep mode Hardware standby mode Module standby mode
Clock pulse generator (CPG)	• On-chip oscillation circuit (Maximum operating frequency: 200 MHz) • Two types of clock generation Internal clock (200 MHz max) Peripheral clock (20 or 40 MHz) • Internal/peripheral clocks multiplied by the on-chip PLL ×4, ×6, ×8, or ×10 can be selected for the internal clock multiplication by setting the MD_CLK0/1 pins. ×1 or ×2 can be selected for the peripheral clock multiplication by setting the MD_CLKP pin. • Input clock frequency: 16 to 20 MHz • 80-MHz clock (for the FlexRay interface module)
Interrupt controller (INTC)	• 9 external interrupt pins (NMI and $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ7}}$) • 15 software interrupts with priority levels 1 to 15 • 16 programmable priority levels

Item	Features
User break controller (UBC)	• 8 channels • Generates an interrupt when the CPU, DMAC, or A-DMAC generates a bus cycle with specified conditions. (Interrupt can be masked.) • Generates a trigger pulse output ($\overline{\text{UBCTRG}}$) on satisfaction of a break condition (the pulse width is selectable as 1, 2, 4, or 8 Pϕ cycles) Pulse width of 1 Pϕ cycle is only available when ×1 multiplication has been set for the peripheral clock
Bus state controller (BSC)	• Supports external memory accesses (SRAM and ROM directly connectable) • 8/16-bit bus • 3.3-V bus interface • 16-Mbyte address space divided into four areas, with the following parameters settable for each area: Bus size: 8 or 16 bits Number of wait cycles Chip select signals ($\overline{\text{CS0}}$ to $\overline{\text{CS3}}$) output for each area • Wait cycle insertion with external $\overline{\text{WAIT}}$ signal • Provision for idle cycle insertion to prevent bus conflicts
Direct memory access controller (DMAC)	• 8 channels • DMA transfer possible between following devices/modules: external memory, on-chip memory, on-chip peripheral modules (excluding DMAC and A-DMAC) • Cycle stealing or burst mode transfer • Dual address mode • Reloading function Reloading source or/and destination address and reload counter Reloading source and destination address, and transfer and reload counter • Integer → floating-point conversion selection during transfer • Transfer data width: byte/word/longword/16 bytes

Item	Features
Automotive direct memory access controller (A-DMAC)	• 78 channels • Forwarding data from/to the specific module to/from alias (specific RAM address) is possible triggered by event generation • Following modules are supported 1 channel for ADC: A/D converted value transfer 1 channel for ATU-III (timer G): RAM data transfer from RAM to PORT Channels for ATU-III (timers A, C, and F): input capture value transfer for timer A (6 channels), timer C (20 channels), and timer F (28 channels) 6 channels for RSPI: MISOA to MISOC and MOSIA to MOSIC 10 channels for SCI: RxD_A to RxD_E and TxD_A to TxD_E 4 channels for RCAN-TL1: CRx_A to CRx_C, CRx_C and CRx_D, CTx_A to CTx_C, and CTx_C and CTx_D 2 channels for FlexRay: FRRxD_A and FRRxD_B • Alias address pointers 1 base pointer: Addresses of aliases for ATU-III (timers A, C, and F), RCAN-TL1, and FlexRay channels relative to the base address are fixed. 18 alias pointers: One for each of ATU-III (timer G), ADC, RSPI, and SCI channels
Advanced timer unit (ATU-III)	• Timer A: 32-bit input capture inputs × 6 channels • Timer B: Angle clock generation timer × 1 channel • Timer C: 24-bit input capture/output compare × 20 channels • Timer D: 24-bit one shot pulse × 16 channels 24-bit output compare × 16 channels • Timer E: 16-bit PWM × 28 channels Each channel has a prescaler of 1/1 to 1/256 scaling. • Timer F: 24-bit event counter × 28 channels • Timer G: 16-bit interval timer × 6 channels • Timer H: 32-bit interval timer × 1 channel • Timer J: 16-bit input capture with 9-stage FIFO × 2 channels • Two external clocks can be input as clock source • External input pins with noise canceller (timer A, C, F, and J) Settings can be made for individual input pins of timer A and F.

Item	Features
Watchdog timer (WDT)	• Selectable from watchdog timer or interval timer function • Internal reset, external signal, or interrupt generated by counter overflow The time taken until the counter overflows is fixed regardless of the multiplication ratio of the peripheral clock signals (20 or 40 MHz). • Power-on reset
Compare match timer (CMT)	• 2 channels • Selection of four counter-input clocks • A compare-match interrupt can be requested independently for each channel
Serial communications interface (SCI)	• 5 channels • Selection of asynchronous or synchronous mode • Simultaneous transmission/reception (full-duplex) capability • Maximum baud rate Asynchronous: 2.5 Mbps • Synchronous: 2.5 Mbps (external clock input) /5 Mbps (internal clock output)
Renesas serial peripheral interface (RSPI)	• 3 channels • Synchronous serial communications • Master/slave mode supported • Programmable bit length, clock polarity, and clock phase • Sequential loop transfer capable • MSB/LSB first selectable • Maximum transfer rate: 10 MHz • Channel A can control up to 8 slaves in single master mode (depends on PFC settings). Channel A can control up to 7 slaves in multi master mode (depends on PFC settings). • Channel B can control up to 4 slaves in single master mode (depends on PFC settings). • Channel C can control up to 4 slaves in single master mode (depends on PFC settings). • Channels B and C can control up to 3 slaves in multi-master mode (depends on PFC settings).

Item	Features
Controller area network (RCAN-TL1)	• 4 channels • TTCAN level 1 support for all channels • BOSCH 2.0B active compatible • Buffer size: transmit/receive × 31, receive only × 1 • Two or more RCAN-TL1 channels can be assigned to one bus to increase number of buffers with a granularity of 32 channels • Parity error detection capability (1-bit parity based; every 8-bit message buffer) • Capability of interrupt generation upon detection of a parity error
FlexRay	• 2 channels (channels A and B which usually transmit or receive the same data) • Conformance with FlexRay protocol specification 2.1 • Buffer size: 8 Kbytes that can be divided to up to 128 units (each configurable as a transmit buffer, receive buffer, or part of the receive FIFO) • Filtering of messages: ID filtering, channel filtering, and cycle-counter filtering • Clock: CPU interface (40 MHz) and protocol engine (80 MHz) • Bite rate: 10 Mbps

Item	Features
A/D converter (ADC)	• 37 channels • 2 sample-and-hold circuits Independent operation of 12-bit 28 channels and 9 channels • Selectable from three conversion modes Continuous scan mode Single scan mode A/D conversion value addition mode (a single channel is converted continuously two to four times) • Conversion trigger capability Scanning can be started by external trigger or ATU-III compare-match • Self-test capability • 0 × Avref, 0.25 × Avref, 0.5 × AVref, 0.75 × Avref, or 1 × AVref voltage (±8 LSB) can be generated internally. • Conversion time: 25/50 Pφ cycles when the peripheral clock multiplication ratio is set to ×1 (Pφ = 20 MHz) 50 Pφ cycles when the peripheral clock multiplication ratio is set to ×2 (Pφ = 40 MHz) • Accuracy: ±8 LSB, Non linearity error: ±4 LSB
JTAG interface	• JTAG port Boundary scan test ports supporting IEEE 1149.1
Advanced user debugger II (AUD-II)	• Eight dedicated pins • RAM monitor mode Data input/output frequency: equal to or lower than both the peripheral clock frequency (Pφ) and 20 MHz Possible to read from or write to a module connected to the internal/external bus • AUD trace mode
I/O ports	• Selectable output driving ability for specific I/O pins • Output inversion enabled or disabled for specific I/O pins • Switchable pull-down resistors for the MISO and RxD pins • CK pin output enabled or disabled by register setting • Edge detection registers provided for specific ports

Item	Features
Multi-input signature generator (MISG)	• Monitors CPU write accesses to specific addresses and generates 32-bit signature by the written data. Signature is generated based on two different polynomials.
ROM	• 4.0-Mbyte flash memory • ROM cache Instruction cache: full associative, 8 lines, 16 byte/line Data cache: full associative, 4 lines, 16 byte/line Line size: 16 byte/line
Intelligent flash security (IFS)	• Protection of the internal ROM against programming and erasing Security level 1: Debugging interface is available. Security level 2: Debugging interface is not available.
EEPROM	• 128-Kbyte EEPROM • Sixteen 8-Kbyte blocks • Writing is possible in 8- or 128-byte units, erasure is possible in 8-Kbyte units
RAM	• 256-Kbyte SRAM • ECC: 1-bit correction and 2-bit detection • ECC can be enabled or disabled
Package	• 272-pin BGA

1.2 Block Diagram

Figure 1.1 shows a block diagram of the SH7256 group.

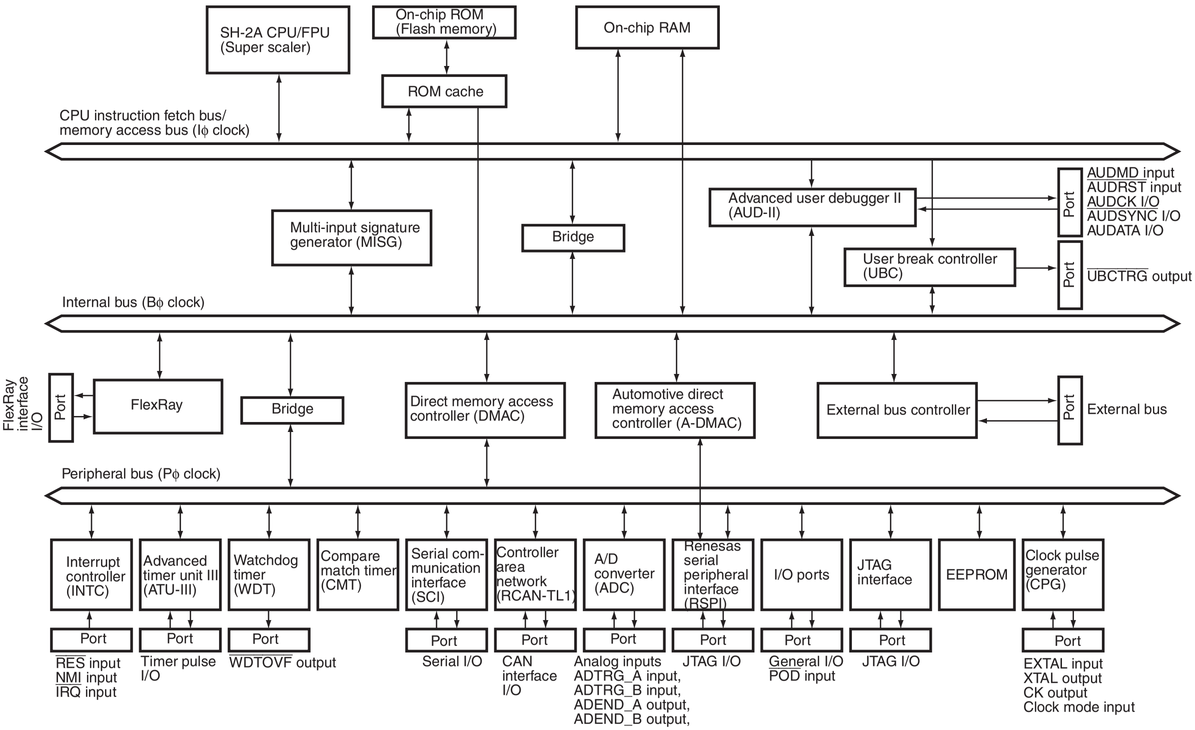
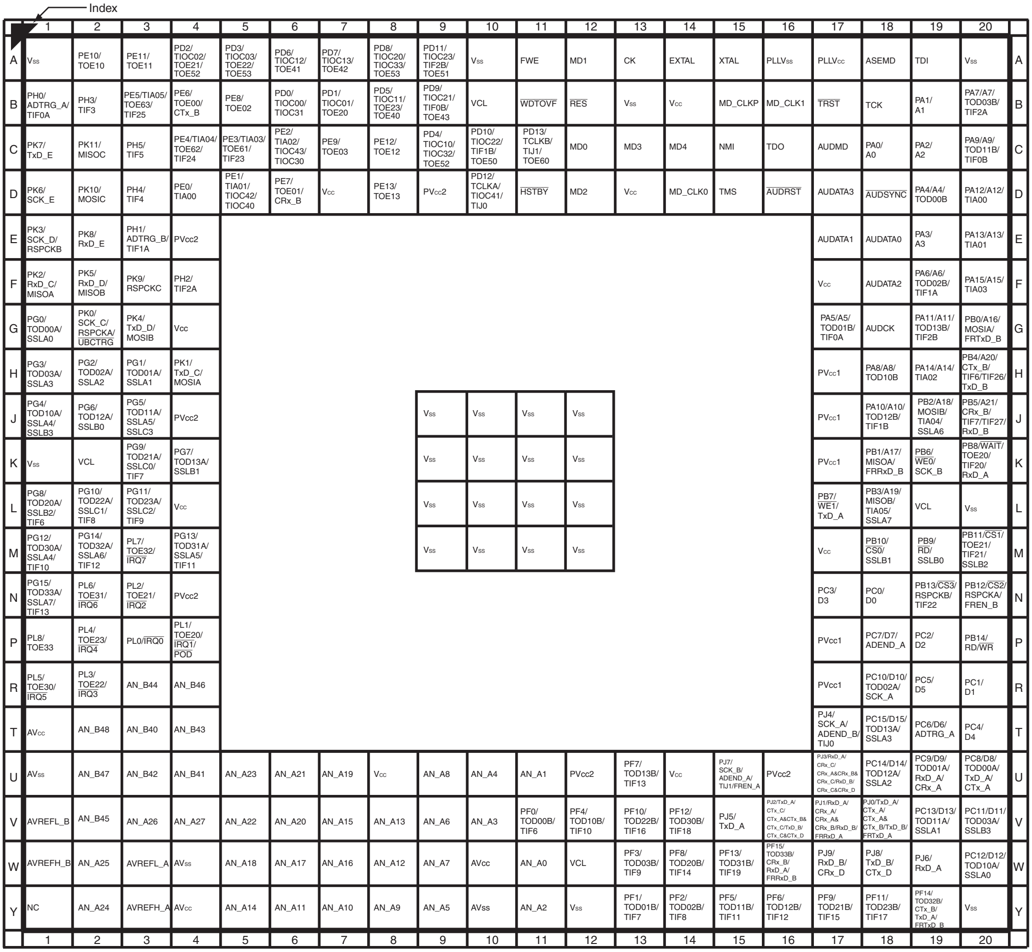


Figure 1.1 Block Diagram

1.3 Pin Descriptions

1.3.1 Pin Assignments

The pin assignments of this LSI is shown in figure 1.2.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	V$_{SS}$	PE10/TOE10	PE11/TOE11	PD2/TIOC02/TOE21/TOE52	PD3/TIOC03/TOE22/TOE53	PD6/TIOC12/TOE41	PD7/TIOC13/TOE42	PD8/TIOC20/TIOC33/TOE53	PD11/TIOC23/TIF2B/TOE51	V$_{SS}$	FWE	MD1	CK	EXTAL	XTAL	PLLV$_{SS}$	PLLV$_{CC}$	ASEMD	TDI	V$_{SS}$
B	PH0/ADTRG_A/TIF0A	PH3/TIF3	PE5/TIA05/TOE63/TIF25	PE6/TOE00/CTx_B	PE8/TOE02	PD0/TIOC00/TIOC31	PD1/TIOC01/TOE20	PD5/TIOC11/TOE23/TOE40	PD9/TIOC21/TIF0B/TOE43	VCL	$\overline{\text{WDTOVF}}$	$\overline{\text{RES}}$	V$_{SS}$	V$_{CC}$	MD_CLKP	MD_CLK1	$\overline{\text{TRST}}$	TCK	PA1/A1	PA7/A7/TOD03B/TIF2A
C	PK7/TxD_E	PK11/MISOC	PH5/TIF5	PE4/TIA04/TOE62/TIF24	PE3/TIA03/TOE61/TIF23	PE2/TIA02/TIOC43/TIOC30	PE9/TOE03	PE12/TOE12	PD4/TIOC10/TIOC32/TOE52	PD10/TIOC22/TIF1B/TOE50	PD13/TCLKB/TIJ1/TOE60	MD0	MD3	MD4	NMI	TDO	AUDMD	PA0/A0	PA2/A2	PA9/A9/TOD11B/TIF0B
D	PK6/SCK_E	PK10/MOSIC	PH4/TIF4	PE0/TIA00	PE1/TIA01/TIOC42/TIOC40	PE7/TOE01/CRx_B	V$_{CC}$	PE13/TOE13	PV$_{CC}$2	PD12/TCLKA/TIOC41/TIJ0	$\overline{\text{HSTBY}}$	MD2	V$_{CC}$	MD_CLK0	TMS	$\overline{\text{AUDRST}}$	AUDATA3	$\overline{\text{AUDSYNC}}$	PA4/A4/TOD00B	PA12/A12/TIA00
E	PK3/SCK_D/RSPCKB	PK8/RxD_E	PH1/ADTRG_B/TIF1A	PVcc2													AUDATA1	AUDATA0	PA3/A3	PA13/A13/TIA01
F	PK2/RxD_C/MISOA	PK5/RxD_D/MISOB	PK9/RSPCKC	PH2/TIF2A													V$_{CC}$	AUDATA2	PA6/A6/TOD02B/TIF1A	PA15/A15/TIA03
G	PG0/TOD00A/SSLA0	PK0/SCK_C/RSPCKA/$\overline{\text{UBCTRG}}$	PK4/TxD_D/MOSIB	Vcc													PA5/A5/TOD01B/TIF0A	AUDCK	PA11/A11/TOD13B/TIF2B	PB0/A16/MOSIA/FRTxD_B
H	PG3/TOD03A/SSLA3	PG2/TOD02A/SSLA2	PG1/TOD01A/SSLA1	PK1/TxD_C/MOSIA													PV$_{CC}$1	PA8/A8/TOD10B	PA14/A14/TIA02	PB4/A20/CTx_B/TIF6/TIF26/TxD_B
J	PG4/TOD10A/SSLA4/SSLB3	PG6/TOD12A/SSLB0	PG5/TOD11A/SSLA5/SSLC3	PVcc2					V$_{SS}$	V$_{SS}$	V$_{SS}$	V$_{SS}$					PV$_{CC}$1	PA10/A10/TOD12B/TIF1B	PB2/A18/MOSIB/TIA04/SSLA6	PB5/A21/CRx_B/TIF7/TIF27/RxD_B
K	V$_{SS}$	VCL	PG9/TOD21A/SSLC0/TIF7	PG7/TOD13A/SSLB1					V$_{SS}$	V$_{SS}$	V$_{SS}$	V$_{SS}$					PV$_{CC}$1	PB1/A17/MISOA/FRRxD_B	PB6/$\overline{\text{WE0}}$/SCK_B	PB8/$\overline{\text{WAIT}}$/TOE20/TIF20/RxD_A
L	PG8/TOD20A/SSLB2/TIF6	PG10/TOD22A/SSLC1/TIF8	PG11/TOD23A/SSLC2/TIF9	V$_{CC}$					V$_{SS}$	V$_{SS}$	V$_{SS}$	V$_{SS}$					PB7/$\overline{\text{WE1}}$/TxD_A	PB3/A19/MISOB/TIA05/SSLA7	VCL	V$_{SS}$
M	PG12/TOD30A/SSLA4/TIF10	PG14/TOD32A/SSLA6/TIF12	PL7/TOE32/$\overline{\text{IRQ7}}$	PG13/TOD31A/SSLA5/TIF11					V$_{SS}$	V$_{SS}$	V$_{SS}$	V$_{SS}$					V$_{CC}$	PB10/$\overline{\text{CS0}}$/SSLB1	PB9/$\overline{\text{RD}}$/SSLB0	PB11/$\overline{\text{CS1}}$/TOE21/TIF21/SSLB2
N	PG15/TOD33A/SSLA7/TIF13	PL6/TOE31/$\overline{\text{IRQ6}}$	PL2/TOE21/$\overline{\text{IRQ2}}$	PVcc2													PC3/D3	PC0/D0	PB13/$\overline{\text{CS3}}$/RSPCKB/TIF22	PB12/$\overline{\text{CS2}}$/RSPCKA/FREN_B
P	PL8/TOE33	PL4/TOE23/$\overline{\text{IRQ4}}$	PL0/$\overline{\text{IRQ0}}$	PL1/TOE20/$\overline{\text{IRQ1}}$/$\overline{\text{POD}}$													PVcc1	PC7/D7/ADEND_A	PC2/D2	PB14/RD/$\overline{\text{WR}}$
R	PL5/TOE30/$\overline{\text{IRQ5}}$	PL3/TOE22/$\overline{\text{IRQ3}}$	AN_B44	AN_B46													PVcc1	PC10/D10/TOD02A/SCK_A	PC5/D5	PC1/D1
T	AV$_{CC}$	AN_B48	AN_B40	AN_B43													PJ4/SCK_A/ADEND_B/TIJ0	PC15/D15/TOD13A/SSLA3	PC6/D6/ADTRG_A	PC4/D4
U	AV$_{SS}$	AN_B47	AN_B42	AN_B41	AN_A23	AN_A21	AN_A19	V$_{CC}$	AN_A8	AN_A4	AN_A1	PVcc2	PF7/TOD13B/TIF13	V$_{CC}$	PJ7/SCK_B/ADEND_A/TIJ1/FREN_A	PVcc2	PJ3/RxD_A/CRx_C/CRx_A&CRx_B&CRx_C/RxD_B/CRx_C&CRx_D	PC14/D14/TOD12A/SSLA2	PC9/D9/TOD01A/RxD_A/CRx_A	PC8/D8/TOD00A/TxD_A/CTx_A
V	AVREFL_B	AN_B45	AN_A26	AN_A27	AN_A22	AN_A20	AN_A15	AN_A13	AN_A6	AN_A3	PF0/TOD00B/TIF6	PF4/TOD10B/TIF10	PF10/TOD22B/TIF16	PF12/TOD30B/TIF18	PJ5/TxD_A	PJ2/TxD_A/CTx_C/CTx_A&CTx_B&CTx_C/TxD_B/CTx_C&CTx_D	PJ1/RxD_A/CRx_A/CRx_A&CRx_B/RxD_B/FRRxD_A	PJ0/TxD_A/CTx_A/CTx_A&CTx_B/TxD_B/FRTxD_A	PC13/D13/TOD11A/SSLA1	PC11/D11/TOD03A/SSLB3
W	AVREFH_B	AN_A25	AVREFL_A	AV$_{SS}$	AN_A18	AN_A17	AN_A16	AN_A12	AN_A7	AVcc	AN_A0	VCL	PF3/TOD03B/TIF9	PF8/TOD20B/TIF14	PF13/TOD31B/TIF19	PF15/TOD33B/CRx_B/RxD_A/FRRxD_B	PJ9/RxD_B/CRx_D	PJ8/TxD_B/CTx_D	PJ6/RxD_A	PC12/D12/TOD10A/SSLA0
Y	NC	AN_A24	AVREFH_A	AV$_{CC}$	AN_A14	AN_A11	AN_A10	AN_A9	AN_A5	AVss	AN_A2	V$_{SS}$	PF1/TOD01B/TIF7	PF2/TOD02B/TIF8	PF5/TOD11B/TIF11	PF6/TOD12B/TIF12	PF9/TOD21B/TIF15	PF11/TOD23B/TIF17	PF14/TOD32B/CTx_B/TxD_A/FRTxD_B	V$_{SS}$

Top View

Figure 1.2 Pin Assignments

1.3.2 Pin Functions

Table 1.2 lists pin functions.

Table 1.2 Pin Functions

Classi- fication	Symbol	Pin No.	I/O	Name	Function
Power supply	V_{CC}	B14, D7, D13, F17, G4, L4, M17, U8, U14	I	Power supply	Power supply pins for internal and system pins. All the V_{CC} pins must be connected to the system power supply. This LSI does not operate correctly if there is a pin left open.
	$PV_{CC}1$	H17, J17, K17, P17, R17	I	5-V/3.3-V power supply (for extended bus)	Power supply for bus ports (ports A, B, and C). Connect all $PV_{CC}1$ pins to the system bus power supply. This LSI does not operate correctly if there is a pin left open.
	$PV_{CC}2$	D9, E4, J4, N4, U12, U16	I	5-V power supply	Power supply for peripheral module ports (ports D, E, F, G, H, J, K, and L). Connect all $PV_{CC}2$ pins to the system peripheral module power supply. This LSI does not operate correctly if there is a pin left open.
	V_{CL}	B10, K2, L19, W12	I	Internal step-down power supply	Pins for connection to a capacitor used for stabilizing the voltage of the internal step-down power supply. Connect V_{SS} to this pin through a capacitor. The capacitor should be located near the pin. Do not connect an external power supply to the pin.

Classi- fication	Symbol	Pin No.	I/O	Name	Function
Power supply	V_{SS}	A1, A10, A20, B13, J9, J10, J11, J12, K1, K9, K10, K11, K12, L9, L10, L11, L12, L20, M9, M10, M11, M12, Y12, Y20	I	Ground	Ground pins. All the V_{SS} pins must be connected to the system power supply (0V). This LSI does not operate correctly if there is a pin left open.
Clocks	$PLLV_{CC}$	A17	I	PLL power supply	Power supply for the on-chip PLL oscillator.
	$PLLV_{SS}$	A16	I	PLL ground	Ground pin for the on-chip PLL oscillator.
	EXTAL	A14	I	External clock	Connected to a crystal resonator. An external clock signal may also be input to the EXTAL pin.
	XTAL	A15	O	Crystal	Connected to a crystal resonator.
	CK	A13	O	Peripheral clock	Supplies the peripheral clock to peripheral devices.
Operating mode control	ASEMD	A18	I	ASE mode	Enables emulator functions. Input a low level to operate the LSI in normal mode (other than debugging mode). In debugging mode, input a high level to this pin on the user system board.
	FWE	A11	I	Flash programming enable	Input a low level in normal operation. Input a high level during on-board programming.
	MD4 to MD0	C12, A12, D12, C13, C14	I	Mode set	Sets the operating mode. Do not change signal levels on these pins during operation.
	MD_CLK1, MD_CLK0, MD_CLKP	D14, B15, B16	I	Clock mode set	Sets the clock operating mode. Do not change signal levels on these pins during operation.

Classification	Symbol	Pin No.	I/O	Name	Function
System control	$\overline{\text{HSTBY}}$	D11	I	Hardware standby	Driving this pin low puts the LSI into hardware standby mode.
	$\overline{\text{RES}}$	B12	I	Power-on reset	Driving this pin low puts the LSI into power-on reset state.
	$\overline{\text{WDTOVF}}$	B11	O	Watchdog timer overflow	WDT overflow output signal
Interrupts	NMI	C15	I	Non-maskable interrupt	Non-maskable interrupt request pin
	$\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$	P3, P4, N3, R2, P2, R1, N2, M3	I	Interrupt requests 7 to 0	Maskable interrupt request pins. Level-input or edge-input detection can be selected. When the edge input detection is selected, the rising edge, falling edge, or both edges can also be selected.
Address bus	A21 to A0	C18, B19 C19, E19, D19, G17, F19, B20, H18, C20, J18, G19, D20, E20, H19, F20, G20, K18, J19, L18, H20, J20	O	Address bus	Address bus

Classi-fication	Symbol	Pin No.	I/O	Name	Function
Data bus	D15 to D0	N18, R20 P19, N17, T20, R19, T19, P18, U20, U19, R18, V20, W20, V19, U18, T18	I/O	Data bus	Bidirectional data bus
Bus control	$\overline{\text{CS3}}$ to $\overline{\text{CS0}}$	M18, M20, N20, N19	O	Chip select 3 to 0	Chip-select signals for external memory or devices
	$\overline{\text{RD}}$	M19	O	Read	Indicates that data is read from an external device.
	RD/$\overline{\text{WR}}$	P20	O	Read/write	Read/write signal.
	$\overline{\text{WAIT}}$	K20	I	Wait control	Input signal for inserting a wait cycle into the bus cycles during access to the external space.
	$\overline{\text{WE0}}$	K19	O	Write/byte select	Indicates a write access to bits 7 to 0 of data bus of external memory or device.
	$\overline{\text{WE1}}$	L17	O	Write/byte select	Indicates a write access to bits 15 to 8 of data bus of external memory or device.
Advanced timer unit II (ATU-III)	TCLKA, TCLKB	D10, C11	I	ATU-III timer clock input	Input pins for the ATU-III counter external clock
	TIA05 to TIA00	D20, E20, H19, F20, J19, L18, D4, D5, C6, C5, C4, B3	I	ATU-III input capture (timer A)	Timer A input capture input pins

Classi- fication	Symbol	Pin No.	I/O	Name	Function
Advanced timer unit II (ATU-III)	TIOC43 to TIOC40, TIOC33 to TIOC30, TIOC23 to TIOC20, TIOC13 to TIOC10, TIOC03 to TIOC00	B6, B7, C10, A9, D10, A4, A5, C9, B8, A6, A7, A8, B9, D5, C6	I/O	ATU-III input capture/ output compare (timer C)	Timer C input capture input/output compare output pins
	TOD33A to TOD30A, TOD23A to TOD20A, TOD13A to TOD10A, TOD03A to TOD00A	U20, U19, R18, V20, W20, V19, U18, T18, G1, H3, L2, L3, M1, M4, M2, N1, H2, H1, J1, J3, J2, K4, L1, K3	O	ATU-III compare match (timer D)	Timer D output compare match output pins
	TOD33B to TOD30B, TOD23B to TOD20B, TOD13B to TOD10B, TOD03B to TOD00B	D19, G17, F19, B20, H18, C20, J18, G19, V11, Y13, V13, Y18, Y14, W15, Y19, W16, Y14, W13, V12, Y15, Y16, U13, W14, Y17	O	ATU-III one-shot pulse (timer D)	Timer D down-counter one-shot pulse output pins

Classi- fication	Symbol	Pin No.	I/O	Name	Function
Advanced timer unit II (ATU-III)	TOE63 to TOE60, TOE53 to TOE50, TOE43 to TOE40, TOE33 to TOE30, TOE23 to TOE20, TOE13 to TOE10, TOE03 to TOE00	B3, C4, C5, C11, B4, D6, B5, C7, A2, A3, C8, D8, P4, K20, B7, N3, M20, A4 R2, A5, P2, B8, R1, N2, M3, P1, A6, A7, B9, C10, A9, C9, A8	O	ATU-III PWM output (timer E)	Timer E output compare/PWM output pins
	TIF2A to TIF0A	G17, F19, B20, B1, E3, F4	I	ATU-III event input (phase A, timer F)	Timer F event input pins (phase A)
	TIF2B to TIF0B	C20, J18, G19, B9, C10, A9	I	ATU-III event input (phase B, timer F)	Timer F event input pins (phase B)

Classi- fication	Symbol	Pin No.	I/O	Name	Function
Advanced timer unit II (ATU-III)	TIF27 to TIF3	K20, M20, N19, C5, C4, B3, V12, M1, Y15, M2, U13, N1, W14, Y17, V13, Y8, V14, W15, B2, D3, C3, V11, H20, L1, Y13, J20, K3, Y14, L2, W13, L3	I	ATU-III event input (timer F)	Timer F event input pins
	TIJ1, TIJ0	D10, T17, C11, U15	I	ATU-III capture input (timer J)	Timer J input capture input pins
Controller area network (RCAN-TL1)	CTx_A to CTx_D	V18, V16, H20, B4, Y19, U20, W18	O	Transmit data (channels A to D)	CAN bus transmit data output pins
	CRx_A to CRx_D	V17, U17, J20, D6, W16, U19, W17	I	Receive data (channels A to D)	CAN bus receive data input pins

Classi- fication	Symbol	Pin No.	I/O	Name	Function
FlexRay	FRTxD_A, FRTxD_B	V18, G20, Y19	O	Transmit data (channels A and B)	FlexRay transmit data output pins
	FREN_A, FREN_B	U15, N20	O	Enable signal (channels A and B)	FlexRay enable signal output pins
	FRRxD_A, FRRxD_B	V17, K18, W16	I	Receive data (channels A and B)	FlexRay receive data input pins
Serial communi-cation interface (SCI)	TxD_A to TxD_E	H20, L17, U20, V18, V16, V15, Y19, W18, H4, G3, C1	O	Transmit data (channels A to E)	SCI_A to SCI_E transmit data output pins
	RxD_A to RxD_E	J20, K20, U19, V17, U17, W19, W16, W17, F1, F2, E2	I	Receive data (channels A to E)	SCI_A to SCI_E receive data input pins
	SCK_A to SCK_E	K19, R18, T17, U15, G2, E1, D1	I/O	Serial clock (channels A to E)	SCI_A to SCI_E clock input/output pins

Classi- fication	Symbol	Pin No.	I/O	Name	Function
Renesas serial peripheral interface (RSPI)	MOSIA to MOSIC	G20, H4, J19, G3, D2	I/O	Transmit data (channels A to C)	RSPI_A to RSPI_C transmit data input/output pins
	MISOA to MISOC	K18, F1, L18, F2, C2	I/O	Receive data (channels A to C)	RSPI_A to RSPI_C receive data input/output pins
	RSPCKA to RSPCKC	N20, G2, N19, E1, F3	I/O	Serial clock (channels A to C)	RSPI_A to RSPI_C clock input/output pins
	SSLA0, SSLB0, SSLC0	M19, W20, G1, J2, K3	I/O	Chip select (channels A to C)	RSPI_A to RSPI_C chip select input/output pins
	SSLA1-7, SSLB1-3, SSLC1-3	J19, L18, M18, M20, V20, V19, U18, T18, H3, H2, H1, M1, J1, M4, J3, M2, N1, K4, L1, L2, L3	O	Chip select (channels A to C)	RSPI_A to RSPI_C chip select input/output pins
User break controller (UBC)	$\overline{\text{UBCTRG}}$	G2	O	User break trigger output	Trigger output pin for UBC condition match
A/D converter (ADC)	AV_{CC}	T1, W10, Y4	I	Analog power supply	Power supply pins for the A/D converter
	AV_{SS}	U1, W4, Y10	I	Analog ground	Ground pins for the A/D converter

Classi-fication	Symbol	Pin No.	I/O	Name	Function
A/D converter (ADC)	AN_A27 to AN_A0, AN_B48 to AN_B40	W11, U11, Y7, Y6, W8, V8, Y5, V7, W7, W6, W5, U7, Y11, V6, U6, V5, U5, Y2, W2, V3, V4, V10, U10, Y9, V9, W9, U9, Y8, T3, U4, U3, T4, R3, V2, R4, U2, T2	I	Analog input pins	Analog input pins
	ADTRG_A, ADTRG_B	T19, B1, E3	I	A/D conversion trigger input	External trigger input pin for starting A/D conversion
	ADEND_A, ADEND_B	P18, U15, T17	O	ADEND output	Output pins for A/D conversion timing monitor
	AVREFH_A, AVREFH_B	Y3, W1	I	Analog reference power supply	Input pins for analog high level reference power supply
	AVREFL_A, AVREFL_B	W3, V1	I	Analog reference power supply	Input pins for analog low level reference power supply
JTAG interface	TCK	B18	I	Test clock	Test-clock input pin
	TMS	D15	I	Test mode select	Test-mode select signal input pin
	TDI	A19	I	Test data input	Serial input pin for instructions and data
	TDO	C16	O	Test data output	Serial output pin for instructions and data
	$\overline{\text{TRST}}$	B17	I	Test reset	Initialization-signal input pin

Classi- fication	Symbol	Pin No.	I/O	Name	Function
Advanced user debugger II (AUD-II)	AUDATA3 to AUDATA0	E18, E17, F18, D17	I/O	AUD data	Branch trace mode: Branch destination address output pins. RAM monitor mode: Monitor address input/data input/output pins.
	$\overline{\text{AUDRST}}$	D16	I	AUD reset	AUD-II reset signal input pin
	AUDMD	C17	I	AUD mode	Mode select signal input pin Branch trace mode: Input a low level RAM monitor mode: Input a high level
	AUDCK	G18	I/O	AUD clock	Branch trace mode: Clock output pin RAM monitor mode: Clock input pin
	$\overline{\text{AUDSYNC}}$	D18	I/O	AUD sync signal	Branch trace mode: Data start position identification signal output pin RAM monitor mode: Data start position identification signal input pin
I/O ports	$\overline{\text{POD}}$	P4	I	Port output disable	Input pin for pin driving control when a general port is set for output
	PA15 to PA0	C18, B19, J18, G19, D20, E20, H19, F20, C19, E19, D19, G17, F19, B20, H18, C20	I/O	Port A	General input/output port pins. Input or output can be specified bit by bit.
	PB14 to PB0	G20, K18, M18, M20, N20, N19, P20, J19, L18, H20, J20, K19, L17, K20, M19	I/O	Port B	General input/output port pins. Input or output can be specified bit by bit.

Classi- fication	Symbol	Pin No.	I/O	Name	Function
I/O ports	PC15 to PC0	N18, R20, R18, V20, W20, V19, U18, T18, P19, N17, T20, R19, T19, P18, U20, U19	I/O	Port C	General input/output port pins. Input or output can be specified bit by bit.
	PD13 to PD0	B6, B7, C10, A9, D10, C11, A4, A5, C9, B8, A6, A7, A8, B9	I/O	Port D	General input/output port pins. Input or output can be specified bit by bit.
	PE13 to PE0	D4, D5, A2, A3, C8, D8, C6, C5, C4, B3, B4, D6, B5, C7	I/O	Port E	General input/output port pins. Input or output can be specified bit by bit.
	PF15 to PF0	V11, Y13, V13, Y18, V14, W15, Y19, W16, Y14, W13, V12, Y15, Y16, U13, W14, Y17	I/O	Port F	General input/output port pins. Input or output can be specified bit by bit.

Classi- fication	Symbol	Pin No.	I/O	Name	Function
I/O ports	PG15 to PG0	G1, H3, L2, L3, M1, M4, M2, N1, H2, H1, J1, J3, J2, K4, L1, K3	I/O	Port G	General input/output port pins. Input or output can be specified bit by bit.
	PH5 to PH0	B1, E3, F4, B2, D3, C3	I/O	Port H	General input/output port pins. Input or output can be specified bit by bit.
	PJ9 to PJ0	V18, V17, V16, U17, T17, V15, W19, U15, W18, W17	I/O	Port J	General input/output port pins. Input or output can be specified bit by bit.
	PK11 to PK0	G2, H4, D2, C2, F1, E1, G3, F2, D1, C1, E2, F3	I/O	Port K	General input/output port pins. Input or output can be specified bit by bit.
	PL8 to PL0	P3, P4, N3, R2, P2, R1, N2, M3, P1	I/O	Port L	General input/output port pins. Input or output can be specified bit by bit.
Other	N.C.	Y1	—	No connection	No connection should be made with this pin.

1.3.3 List of Pins

Table 1.3 List of Pins

Pin Number	MCU Expansion Mode, MCU Single Chip Mode, Boot Mode, User Program Mode, User Boot Mode
A18	ASEMD
C14	MD4
C13	MD3
D12	MD2
A12	MD1
C12	MD0
A11	FWE
B16	MD_CLK1
D14	MD_CLK0
B15	MD_CLKP
D11	$\overline{\text{HSTBY}}$
B12	$\overline{\text{RES}}$
C15	NMI
A14	EXTAL
A15	XTAL
A13	CK
B11	$\overline{\text{WDTOVF}}$
B17	$\overline{\text{TRST}}$
B18	TCK
D15	TMS
A19	TDI
C16	TDO
C17	AUDMD
D16	$\overline{\text{AUDRST}}$
G18	AUDCK
D18	$\overline{\text{AUDSYNC}}$
D17	AUDATA3
F18	AUDATA2

Pin Number	MCU Expansion Mode, MCU Single Chip Mode, Boot Mode, User Program Mode, User Boot Mode
E17	AUDATA1
E18	AUDATA0
C18	PA0/A0
B19	PA1/A1
C19	PA2/A2
E19	PA3/A3
D19	PA4/A4/TOD00B
G17	PA5/A5/TOD01B/TIF0A
F19	PA6/A6/TOD02B/TIF1A
B20	PA7/A7/TOD03B/TIF2A
H18	PA8/A8/TOD10B
C20	PA9/A9/TOD11B/TIF0B
J18	PA10/A10/TOD12B/TIF1B
G19	PA11/A11/TOD13B/TIF2B
D20	PA12/A12/TIA00
E20	PA13/A13/TIA01
H19	PA14/A14/TIA02
F20	PA15/A15/TIA03
G20	PB0/A16/MOSIA/FRTxD_B
K18	PB1/A17/MISOA/FRRxD_B
J19	PB2/A18/MOSIB/TIA04/SSLA6
L18	PB3/A19/MISOB/TIA05/SSLA7
H20	PB4/A20/CTx_B/TIF6/TIF26/TxD_B
J20	PB5/A21/CRx_B/TIF7/TIF27/RxD_B
K19	PB6/$\overline{\text{WE0}}$/SCK_B
L17	PB7/$\overline{\text{WE1}}$/TxD_A
K20	PB8/$\overline{\text{WAIT}}$/TOE20/TIF20/RxD_A
M19	PB9/$\overline{\text{RD}}$/SSLB0
M18	PB10/$\overline{\text{CS0}}$/SSLB1
M20	PB11/$\overline{\text{CS1}}$/TOE21/TIF21/SSLB2
N20	PB12/$\overline{\text{CS2}}$/RSPCKA/FREN_B
N19	PB13/$\overline{\text{CS3}}$/RSPCKB/TIF22

Pin Number	MCU Expansion Mode, MCU Single Chip Mode, Boot Mode, User Program Mode, User Boot Mode
P20	PB14/RD/$\overline{\text{WR}}$
N18	PC0/D0
R20	PC1/D1
P19	PC2/D2
N17	PC3/D3
T20	PC4/D4
R19	PC5/D5
T19	PC6/D6/ADTRG_A
P18	PC7/D7/ADEND_A
U20	PC8/D8/TOD00A/TxD_A/CTx_A
U19	PC9/D9/TOD01A/RxD_A/CRx_A
R18	PC10/D10/TOD02A/SCK_A
V20	PC11/D11/TOD03A/SSLB3
W20	PC12/D12/TOD10A/SSLA0
V19	PC13/D13/TOD11A/SSLA1
U18	PC14/D14/TOD12A/SSLA2
T18	PC15/D15/TOD13A/SSLA3
B6	PD0/TIOC00/TIOC31
B7	PD1/TIOC01/TOE20
A4	PD2/TIOC02/TOE21/TOE52
A5	PD3/TIOC03/TOE22/TOE53
C9	PD4/TIOC10/TIOC32/TOE52
B8	PD5/TIOC11/TOE23/TOE40
A6	PD6/TIOC12/TOE41
A7	PD7/TIOC13/TOE42
A8	PD8/TIOC20/TIOC33/TOE53
B9	PD9/TIOC21/TIF0B/TOE43
C10	PD10/TIOC22/TIF1B/TOE50
A9	PD11/TIOC23/TIF2B/TOE51
D10	PD12/TCLKA/TIOC41/TIJ0

Pin Number	MCU Expansion Mode, MCU Single Chip Mode, Boot Mode, User Program Mode, User Boot Mode
C11	PD13/TCLKB/TIJ1/TOE60
D4	PE0/TIA00
D5	PE1/TIA01/TIOC42/TIOC40
C6	PE2/TIA02/TIOC43/TIOC30
C5	PE3/TIA03/TOE61/TIF23
C4	PE4/TIA04/TOE62/TIF24
B3	PE5/TIA05/TOE63/TIF25
B4	PE6/TOE00/CTx_B
D6	PE7/TOE01/CRx_B
B5	PE8/TOE02
C7	PE9/TOE03
A2	PE10/TOE10
A3	PE11/TOE11
C8	PE12/TOE12
D8	PE13/TOE13
V11	PF0/TOD00B/TIF6
Y13	PF1/TOD01B/TIF7
Y14	PF2/TOD02B/TIF8
W13	PF3/TOD03B/TIF9
V12	PF4/TOD10B/TIF10
Y15	PF5/TOD11B/TIF11
Y16	PF6/TOD12B/TIF12
U13	PF7/TOD13B/TIF13
W14	PF8/TOD20B/TIF14
Y17	PF9/TOD21B/TIF15
V13	PF10/TOD22B/TIF16
Y18	PF11/TOD23B/TIF17
V14	PF12/TOD30B/TIF18
W15	PF13/TOD31B/TIF19
Y19	PF14/TOD32B/CTx_B/TxD_A/FRTxD_B
W16	PF15/TOD33B/CRx_B/RxD_A/FRRxD_B

Pin Number	MCU Expansion Mode, MCU Single Chip Mode, Boot Mode, User Program Mode, User Boot Mode
G1	PG0/TOD00A/SSLA0
H3	PG1/TOD01A/SSLA1
H2	PG2/TOD02A/SSLA2
H1	PG3/TOD03A/SSLA3
J1	PG4/TOD10A/SSLA4/SSLB3
J3	PG5/TOD11A/SSLA5/SSLC3
J2	PG6/TOD12A/SSLB0
K4	PG7/TOD13A/SSLB1
L1	PG8/TOD20A/SSLB2/TIF6
K3	PG9/TOD21A/SSLC0/TIF7
L2	PG10/TOD22A/SSLC1/TIF8
L3	PG11/TOD23A/SSLC2/TIF9
M1	PG12/TOD30A/SSLA4/TIF10
M4	PG13/TOD31A/SSLA5/TIF11
M2	PG14/TOD32A/SSLA6/TIF12
N1	PG15/TOD33A/SSLA7/TIF13
B1	PH0/ADTRG_A/TIF0A
E3	PH1/ADTRG_B/TIF1A
F4	PH2/TIF2A
B2	PH3/TIF3
D3	PH4/TIF4
C3	PH5/TIF5
V18	PJ0/TxD_A/CTx_A/CTx_A&CTx_B/TxD_B/FRTxD_A
V17	PJ1/RxD_A/CRx_A/CRx_A&CRx_B/RxD_B/FRRxD_A
V16	PJ2/TxD_A/CTx_C/CTx_A&CTx_B&CTx_C/TxD_B/CTx_C&CTx_D
U17	PJ3/RxD_A/CRx_C/CRx_A&CRx_B&CRx_C/RxD_B/CRx_C&CRx_D
T17	PJ4/SCK_A/ADEND_B/TIJ0
V15	PJ5/TxD_A
W19	PJ6/RxD_A
U15	PJ7/SCK_B/ADEND_A/TIJ1/FREN_A
W18	PJ8/TxD_B/CTx_D
W17	PJ9/RxD_B/CRx_D

Pin Number	MCU Expansion Mode, MCU Single Chip Mode, Boot Mode, User Program Mode, User Boot Mode
G2	PK0/SCK_C/RSPCKA/$\overline{\text{UBCTRG}}$
H4	PK1/TxD_C/MOSIA
F1	PK2/RxD_C/MISOA
E1	PK3/SCK_D/RSPCKB
G3	PK4/TxD_D/MOSIB
F2	PK5/RxD_D/MISOB
D1	PK6/SCK_E
C1	PK7/TxD_E
E2	PK8/RxD_E
F3	PK9/RSPCKC
D2	PK10/MOSIC
C2	PK11/MISOC
P3	PL0/$\overline{\text{IRQ0}}$
P4	PL1/TOE20/$\overline{\text{IRQ1}}$/$\overline{\text{POD}}$
N3	PL2/TOE21/$\overline{\text{IRQ2}}$
R2	PL3/TOE22/$\overline{\text{IRQ3}}$
P2	PL4/TOE23/$\overline{\text{IRQ4}}$
R1	PL5/TOE30/$\overline{\text{IRQ5}}$
N2	PL6/TOE31/$\overline{\text{IRQ6}}$
M3	PL7/TOE32/$\overline{\text{IRQ7}}$
P1	PL8/TOE33
W11	AN_A0
U11	AN_A1
Y11	AN_A2
V10	AN_A3
U10	AN_A4
Y9	AN_A5
V9	AN_A6
W9	AN_A7
U9	AN_A8
Y8	AN_A9

Pin Number	MCU Expansion Mode, MCU Single Chip Mode, Boot Mode, User Program Mode, User Boot Mode
Y7	AN_A10
Y6	AN_A11
W8	AN_A12
V8	AN_A13
Y5	AN_A14
V7	AN_A15
W7	AN_A16
W6	AN_A17
W5	AN_A18
U7	AN_A19
V6	AN_A20
U6	AN_A21
V5	AN_A22
U5	AN_A23
Y2	AN_A24
W2	AN_A25
V3	AN_A26
V4	AN_A27
Y3	AVREFH_A
W3	AVREFL_A
T3	AN_B40
U4	AN_B41
U3	AN_B42
T4	AN_B43
R3	AN_B44
V2	AN_B45
R4	AN_B46
U2	AN_B47
T2	AN_B48
W1	AVREFH_B

Pin Number	MCU Expansion Mode, MCU Single Chip Mode, Boot Mode, User Program Mode, User Boot Mode
V1	AVREFL_B
A17	$PLLV_{CC}$
A16	$PLLV_{SS}$
B10	V_{CL}
K2	V_{CL}
L19	V_{CL}
W12	V_{CL}
B14	V_{CC}
D7	V_{CC}
D13	V_{CC}
F17	V_{CC}
G4	V_{CC}
L4	V_{CC}
M17	V_{CC}
U8	V_{CC}
U14	V_{CC}
H17	$PV_{CC}1$
J17	$PV_{CC}1$
K17	$PV_{CC}1$
P17	$PV_{CC}1$
R17	$PV_{CC}1$
D9	$PV_{CC}2$
E4	$PV_{CC}2$
J4	$PV_{CC}2$
N4	$PV_{CC}2$
U12	$PV_{CC}2$
U16	$PV_{CC}2$
A1	V_{SS}
A10	V_{SS}
A20	V_{SS}
B13	V_{SS}

Pin Number	MCU Expansion Mode, MCU Single Chip Mode, Boot Mode, User Program Mode, User Boot Mode
J9	V_{SS}
J10	V_{SS}
J11	V_{SS}
J12	V_{SS}
K1	V_{SS}
K9	V_{SS}
K10	V_{SS}
K11	V_{SS}
K12	V_{SS}
L9	V_{SS}
L10	V_{SS}
L11	V_{SS}
L12	V_{SS}
L20	V_{SS}
M9	V_{SS}
M10	V_{SS}
M11	V_{SS}
M12	V_{SS}
Y12	V_{SS}
Y20	V_{SS}
T1	AV_{CC}
W10	AV_{CC}
Y4	AV_{CC}
U1	AV_{SS}
W4	AV_{SS}
Y10	AV_{SS}
Y1	NC

Section 2 CPU

2.1 Data Format

Figure 2.1 shows the data format supported by the SH-2A/SH2A-FPU.

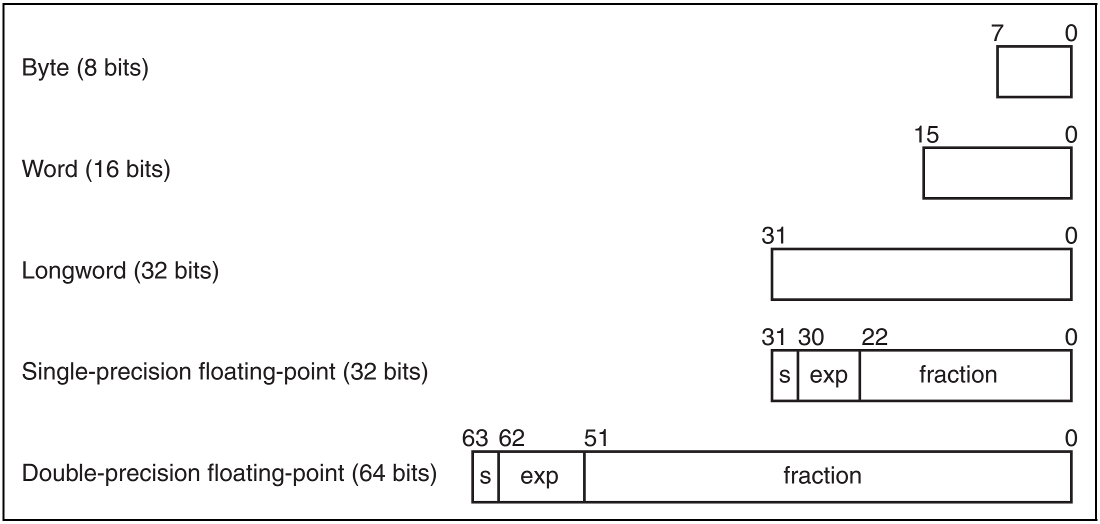


Figure 2.1 Data Format

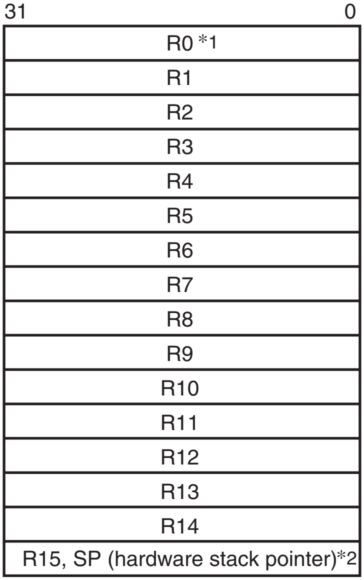
2.2 Register Descriptions

The register set consists of five types of registers: sixteen 32-bit general registers, four 32-bit control registers, four 32-bit system registers, floating-point registers, and floating-point system registers.

2.2.1 General Registers

Figure 2.2 shows the general registers.

The general registers consists of 16 registers, numbered R0 to R15, and are used for data processing and address calculation. R0 is also used as an index register. Several instructions have R0 fixed as their only usable register. R15 is used as the hardware stack pointer (SP). Saving and restoring the status register (SR) and program counter (PC) in exception handling is accomplished by referencing the stack using R15.



Notes: 1. R0 functions as an indexed register in the indexed register indirect and indexed GBR indirect addressing modes. Several instructions have R0 fixed as their source or destination register.
2. R15 is used as a hardware stack pointer (SP) in exception handling.

Figure 2.2 General Registers

2.2.2 Control Registers

The control registers consist of four 32-bit registers: the status register (SR), the global base register (GBR), the vector base register (VBR), and the jump table base register (TBR).

The status register indicates instruction processing states.

The global base register functions as a base address for the GBR indirect addressing mode to transfer data to the registers of on-chip peripheral modules.

The vector base register functions as the base address of the exception handling vector area (including interrupts).

The jump table base register functions as the base address of the function table area.

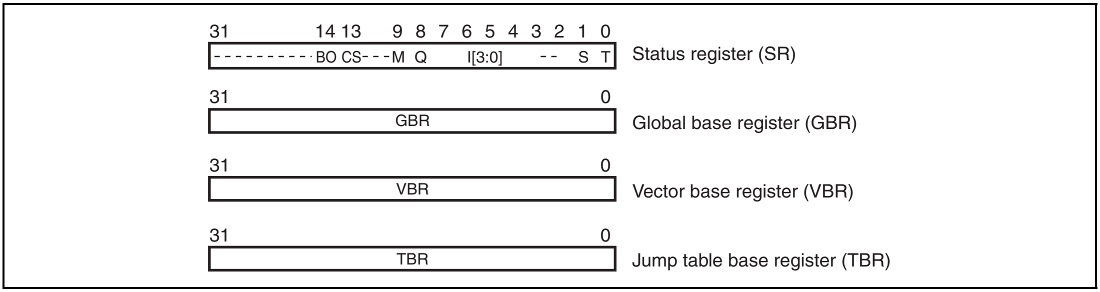

31
14 13
9 8 7 6 5 4 3 2 1 0
- - - - - - - - - BO CS - - - M Q I[3:0] - - S T
Status register (SR)
31
0
GBR
Global base register (GBR)
31
0
VBR
Vector base register (VBR)
31
0
TBR
Jump table base register (TBR)

Figure 2.3 Control Registers

(1) Status Register (SR)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	BO	CS	-	-	-	M	Q	I[3:0]				-	-	S	T
Initial value:	0	0	0	0	0	0	-	-	1	1	1	1	0	0	-	-
R/W:	R	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 15	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
14	BO	0	R/W	BO Bit Indicates the register bank has overflowed.
13	CS	0	R/W	CS Bit Indicates, in CLIP instruction execution, the value has exceeded the saturation upper-limit value or fallen below the saturation lower-limit value.
12 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

		Initial		
Bit	Bit Name	Value	R/W	Description
9	M	—	R/W	M Bit
8	Q	—	R/W	Q Bit Used by the DIV0S, DIV0U, and DIV1 instructions.
7 to 4	I[3:0]	1111	R/W	Interrupt Mask Level
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	S	—	R/W	S Bit Specifies a saturation operation for a MAC instruction.
0	T	—	R/W	T Bit True/false condition or carry/borrow bit

(2) Global Base Register (GBR)

GBR is referenced as the base address in a GBR-referencing MOV instruction.

(3) Vector Base Register (VBR)

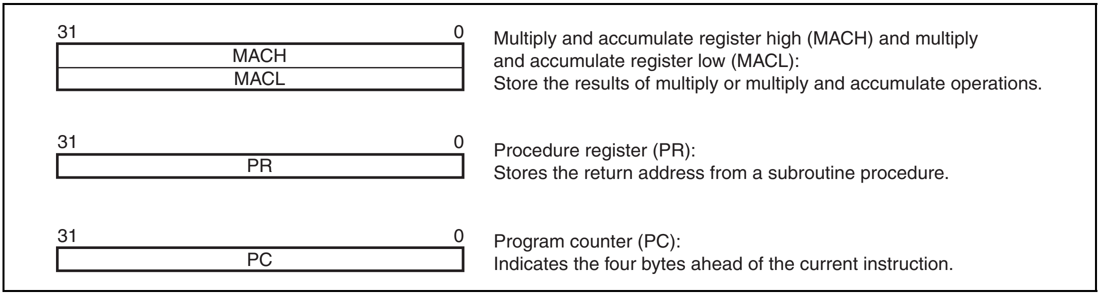
VBR is referenced as the branch destination base address when an exception or an interrupt occurs.

(4) Jump Table Base Register (TBR)

TBR is referenced as the start address of a function table located in memory in a JSR/N@@(disp8,TBR) table-referencing subroutine call instruction.

2.2.3 System Registers

The system registers consist of four 32-bit registers: the high and low multiply and accumulate registers (MACH and MACL), the procedure register (PR), and the program counter (PC). MACH and MACL store the results of multiply or multiply and accumulate operations. PR stores the return address from a subroutine procedure. PC indicates the program address being executed and controls the flow of the processing.



Multiply and accumulate register high (MACH) and multiply and accumulate register low (MACL):
Store the results of multiply or multiply and accumulate operations.

Procedure register (PR):
Stores the return address from a subroutine procedure.

Program counter (PC):
Indicates the four bytes ahead of the current instruction.

Figure 2.4 System Registers

(1) Multiply and Accumulate Register High (MACH) and Multiply and Accumulate Register Low (MACL)

MACH and MACL are used as the addition value in a MAC instruction, and store the result of a MAC or MUL instruction.

(2) Procedure Register (PR)

PR stores the return address of a subroutine call using a BSR, BSRF, or JSR instruction, and is referenced by a subroutine return instruction (RTS).

(3) Program Counter (PC)

PC indicates the address four bytes ahead of the instruction being currently executed.

2.2.4 Floating-Point Registers

Figure 2.5 shows the floating-point registers. There are sixteen 32-bit floating-point registers, FPR0 to FPR15. These sixteen registers are referenced as FR0 to FR15, DR0, DR2, DR4, DR6, DR8, DR10, DR12, and DR14. The correspondence between FPRn and the referenced name is determined by the PR and SZ bits in FPSCR (see figure 2.5).

(1) Floating-Point Registers (FPRn: 16 registers)

FPR0, FPR1, FPR2, FPR3, FPR4, FPR5, FPR6, FPR7, FPR8, FPR9, FPR10, FPR11, FPR12, FPR13, FPR14, and FPR15

(2) Single-Precision Floating-Point Registers (FRi: 16 registers)

FR0 to FR15 are allocated to FPR0 to FPR15.

(3) Double-Precision Floating-Point Registers or Single-Precision Floating-Point Register Pairs (DRi: 8 registers)

A DR register is composed of two FR registers.

DR0 = {FPR0, FPR1}, DR2 = {FPR2, FPR3}, DR4 = {FPR4, FPR5}, DR6 = {FPR4, FPR5}, DR8 = {FPR8, FPR9}, DR10 = {FPR10, FPR11}, DR12 = {FPR12, FPR13}, and DR14 = {FPR14, FPR15}

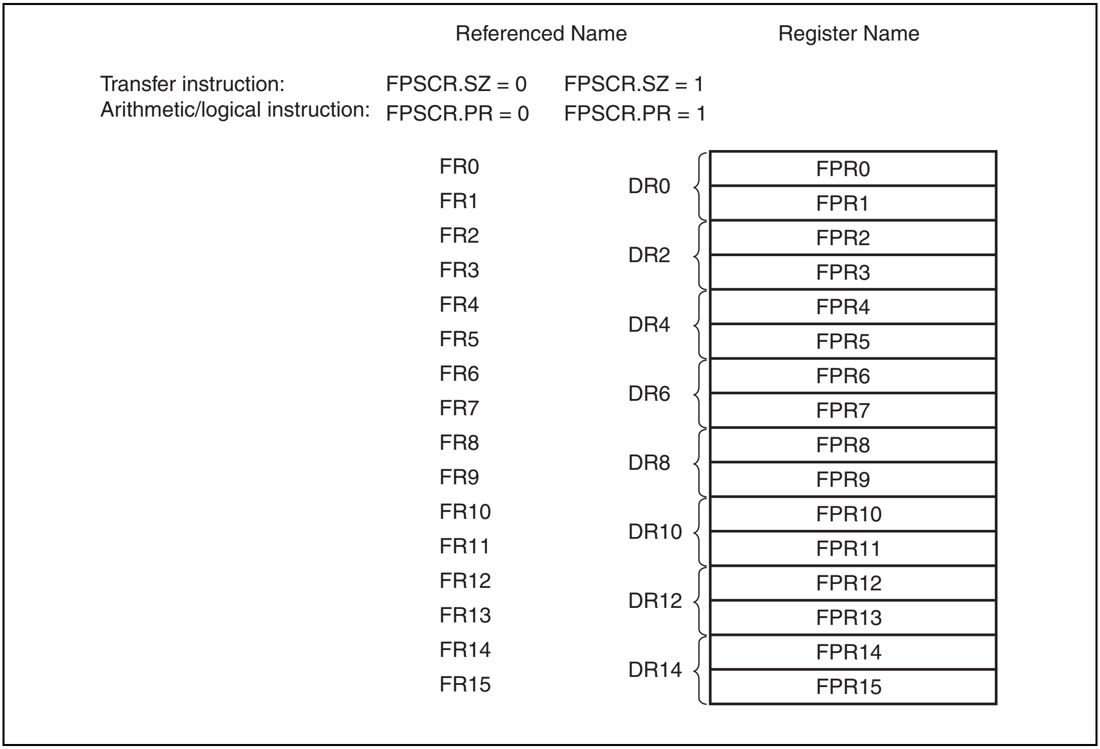


Figure 2.5 Floating-Point Registers

Programming Note: The values of FPR0 to FPR15 are undefined after a reset.

2.2.5 Floating-Point System Registers

(1) Floating-Point Communication Register (FPUL)

Data is transferred between an FPU register and a CPU register via FPUL.

(2) Floating Point Status/Control Register (FPSCR)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	QIS	-	SZ	PR	DN	Cause	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
R/W:	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Cause				Enable					Flag					RM[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 23	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
22	QIS	0	R/W	sNaN is treated as qNaN or ±∞. Valid only when the V bit in the FPU exception enable field (Enable) is set to 1. 0: Processed as qNaN or ±∞ 1: Exception generated (processed same as sNaN)
21	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
20	SZ	0	R/W	Transfer Size Mode 0: Sets the size of an FMOV instruction to 32 bits. 1: Sets the size of an FMOV instruction to 32-bit pair (64 bits).
19	PR	0	R/W	Precision Mode 0: Executes floating-point instructions in single precision. 1: Executes floating-point instructions in double precision (the result of an instruction with no support for double-precision is undefined).

Bit	Bit Name	Initial Value	R/W	Description
18	DN	1	R/W	Denormalization Mode This bit is always set to 1. 1: A denormalized number is treated as zero.
17 to 12	Cause	All 0	R/W	FPU exception cause field
11 to 7	Enable	All 0	R/W	FPU exception enable field
6 to 2	Flag	All 0	R/W	FPU exception flag field When an FPU operation instruction is first executed, the FPU exception cause field is set to 0; when an FPU exception next occurs, the corresponding bit in the FPU exception cause field and FPU exception flag field is set to 1. The FPU exception flag field retains the status of an exception generated after that field was last cleared. For bit allocation for each field, see table 2.1.
1, 0	RM[1:0]	01	R/W	Round Mode 00: Round to nearest 01: Round to zero 10: Reserved 11: Reserved

Table 2.1 Bit Allocation for FPU Exception Handling

		FPU Error (E)	Invalid Operation (V)	Division by 0 (Z)	Overflow (O)	Underflow (U)	Incorrect (I)
Cause	FPU exception cause field	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12
Enable	FPU exception enable field	None	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7
Flag	FPU exception flag field	None	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2

Note: In the SH-2A, no FPU errors occur.

2.2.6 Register Bank

Using a register bank, high-speed register saving and restoration can be achieved for the 19 32-bit registers: general registers R0 to R14, control register GBR, and system registers MACH, MACL, and PR. The register contents are automatically saved in the bank after the CPU accepts an interrupt that uses the bank. Restoration from the bank is executed by a RESBANK instruction issued in an interrupt processing routine.

For details, refer to the SH-2A, SH2A-FPU Software Manual.

2.2.7 Initial Values of Registers

Table 2.2 lists the values of the registers after a reset.

Table 2.2 Initial Values of Registers

Classification	Register	Initial Value
General registers	R0 to R14	Undefined
	R15 (SP)	Value of the stack pointer in the vector address table
Control registers	SR	Bits I[3:0] are 1111 (H'F), BO and CS are 0, reserved bits are 0, and others are undefined
	GBR, TBR	Undefined
	VBR	H'00000000
System registers	MACH, MACL, PR	Undefined
	PC	Value of the program counter in the vector address table
Floating-point registers	FPR0 to FPR15	Undefined
Floating-point system registers	FPUL	Undefined
	FPSCR	H'00040001

2.3 Data Formats

2.3.1 Data Format in Registers

Register operands are always longwords (32 bits). If the size of a memory operand is a byte (8 bits) or a word (16 bits), it is changed into a longword through sign extension or zero extension when loaded into a register.

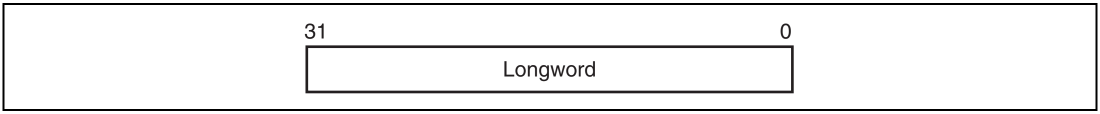


Figure 2.6 Data Format in Registers

2.3.2 Data Formats in Memory

Memory data formats are classified into bytes, words, and longwords. Memory can be accessed in 8-bit bytes, 16-bit words, or 32-bit longwords. A memory operand of fewer than 32 bits is stored in a register in sign-extended or zero-extended form.

A word operand should be accessed at a word boundary (an even address of multiple of two bytes: address 2n), and a longword operand at a longword boundary (an even address of multiple of four bytes: address 4n). Otherwise, an address error will occur. A byte operand can be accessed at any address.

Only big-endian byte order can be selected for the data format.

Data formats in memory are shown in figure 2.7.

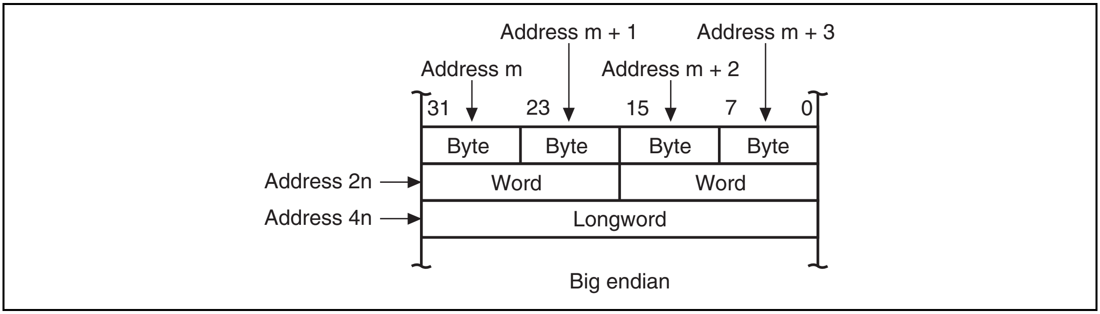


Figure 2.7 Data Formats in Memory

2.3.3 Immediate Data Format

Byte (8-bit) immediate data is located in an instruction code. Immediate data accessed by the MOV, ADD, and CMP/EQ instructions is sign-extended and handled in registers as longword data. Immediate data accessed by the TST, AND, OR, and XOR instructions is zero-extended and handled as longword data. Consequently, AND instructions with immediate data always clear the upper 24 bits of the destination register.

20-bit immediate data is located in the code of a MOVI20 or MOVI20S 32-bit transfer instruction. The MOVI20 instruction stores immediate data in the destination register in sign-extended form. The MOVI20S instruction shifts immediate data by eight bits in the upper direction, and stores it in the destination register in sign-extended form.

Word or longword immediate data is not located in the instruction code, but rather is stored in a memory table. The memory table is accessed by an immediate data transfer instruction (MOV) using the PC relative addressing mode with displacement.

See examples given in section 2.4.1 (10), Immediate Data.

2.4 Instruction Features

2.4.1 RISC-Type Instruction Set

The CPU has a RISC-type instruction set, which features following functions.

(1) 16-Bit Fixed-Length Instructions

Basic instructions have a fixed length of 16 bits, improving program code efficiency.

(2) 32-Bit Fixed-Length Instructions

The SH-2A/SH2A-FPU additionally features 32-bit fixed-length instructions, improving performance and ease of use.

(3) One Instruction per Cycle

Each basic instruction can be executed in one cycle using the pipeline system.

(4) Data Length

The standard data length for all operations is a longword. Memory can be accessed in bytes, words, or longwords. Byte or word data in memory is sign-extended and handled as longword data. Immediate data is sign-extended for arithmetic operations or zero-extended for logic operations. It is also handled as longword data.

Table 2.3 Sign Extension of Word Data

SH2-A/SH2A-FPU CPU		Description	Example of Other CPU
MOV.W	@(disp,PC),R1	Data is sign-extended to 32 bits, and R1 becomes H'00001234. It is next operated upon by an ADD instruction.	ADD.W #H'1234,R0
ADD	R1,R0		
....			
.DATA.W	H'1234		

Note: @(disp, PC) accesses the immediate data.

(5) Load-Store Architecture

Basic operations are executed between registers. For operations that involve memory access, data is loaded to the registers and executed (load-store architecture). Instructions such as AND that manipulate bits, however, are executed directly in memory.

(6) Delayed Branch Instructions

With the exception of some instructions, unconditional branch instructions, etc., are executed as delayed branch instructions. With a delayed branch instruction, the branch is taken after execution of the instruction immediately following the delayed branch instruction. This reduces disturbance of the pipeline control when a branch is taken.

In a delayed branch, the actual branch operation occurs after execution of the slot instruction. However, instruction execution such as register updating excluding the actual branch operation, is performed in the order of delayed branch instruction → delay slot instruction. For example, even though the contents of the register holding the branch destination address are changed in the delay slot, the branch destination address remains as the register contents prior to the change.

Table 2.4 Delayed Branch Instructions

SH2-A/SH2A-FPU CPU		Description	Example of Other CPU	
`BRA`	`TRGET`	Executes the ADD before branching to TRGET.	`ADD.W`	`R1,R0`
`ADD`	`R1,R0`		`BRA`	`TRGET`

(7) Unconditional Branch Instructions with No Delay Slot

The SH-2A/SH2A-FPU additionally features unconditional branch instructions in which a delay slot instruction is not executed. This eliminates unnecessary NOP instructions, and so reduces the code size.

(8) Multiply/Multiply-and-Accumulate Operations

16-bit × 16-bit → 32-bit multiply operations are executed in one to two cycles. 16-bit × 16-bit + 64-bit → 64-bit multiply-and-accumulate operations are executed in two to three cycles. 32-bit × 32-bit → 64-bit multiply and 32-bit × 32-bit + 64-bit → 64-bit multiply-and-accumulate operations are executed in two to four cycles.

(9) T Bit

The T bit in the status register (SR) changes according to the result of the comparison. Whether a conditional branch is taken or not taken depends upon the T bit condition (true/false). The number of instructions that change the T bit is kept to a minimum to improve the processing speed.

Table 2.5 T Bit

SH2-A/SH2A-FPU CPU		Description	Example of Other CPU	
CMP/GE	R1,R0	T bit is set when R0 ≥ R1.	CMP.W	R1,R0
BT	TRGET0	The program branches to TRGET0 when R0 ≥ R1 and to TRGET1 when R0 < R1.	BGE	TRGET0
BF	TRGET1		BLT	TRGET1
ADD	#−1,R0	T bit is not changed by ADD.	SUB.W	#1,R0
CMP/EQ	#0,R0	T bit is set when R0 = 0.	BEQ	TRGET
BT	TRGET	The program branches if R0 = 0.		

(10) Immediate Data

Byte immediate data is located in an instruction code. Word or longword immediate data is not located in instruction codes but in a memory table. The memory table is accessed by an immediate data transfer instruction (MOV) using the PC relative addressing mode with displacement.

With the SH-2A/SH2A-FPU, 17- to 28-bit immediate data can be located in an instruction code. However, for 21- to 28-bit immediate data, an OR instruction must be executed after the data is transferred to a register.

Table 2.6 Immediate Data Accessing

Classification	SH-2A/SH2A-FPU CPU		Example of Other CPU	
8-bit immediate	MOV	#H'12,R0	MOV.B	#H'12,R0
16-bit immediate	MOVI20	#H'1234,R0	MOV.W	#H'1234,R0
20-bit immediate	MOVI20	#H'12345,R0	MOV.L	#H'12345,R0
28-bit immediate	MOVI20S OR	#H'12345,R0 #H'67,R0	MOV.L	#H'1234567,R0
32-bit immediate	MOV.L .... .DATA.L	@(disp,PC),R0 H'12345678	MOV.L	#H'12345678,R0

Note: @(disp, PC) accesses the immediate data.

(11) Absolute Address

When data is accessed by an absolute address, the absolute address value should be placed in the memory table in advance. That value is transferred to the register by loading the immediate data during the execution of the instruction, and the data is accessed in register indirect addressing mode.

With the SH-2A/SH2A-FPU, when data is referenced using an absolute address not exceeding 28 bits, it is also possible to transfer immediate data located in the instruction code to a register and to reference the data in register indirect addressing mode. However, when referencing data using an absolute address of 21 to 28 bits, an OR instruction must be used after the data is transferred to a register.

Table 2.7 Absolute Address Accessing

Classification	SH-2A/SH2A-FPU CPU	Example of Other CPU

Up to 20 bits	`MOVI20 #H'12345,R1`
	`MOV.B @R1,R0`
		`MOV.B @H'12345,R0`

21 to 28 bits	`MOVI20S #H'12345,R1`
	`OR #H'67,R1`
	`MOV.B @R1,R0`
		`MOV.B @H'1234567,R0`

29 bits or more	`MOV.L @(disp,PC),R1`
	`MOV.B @R1,R0`
	`..................`
	`.DATA.L H'12345678`
		`MOV.B @H'12345678,R0`

(12) 16-Bit/32-Bit Displacement

When data is accessed by 16-bit or 32-bit displacement, the displacement value should be placed in the memory table in advance. That value is transferred to the register by loading the immediate data during the execution of the instruction, and the data is accessed in the indexed indirect register addressing mode.

Table 2.8 Displacement Accessing

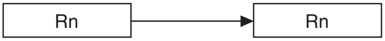
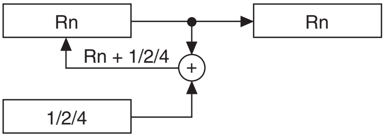
Classification	SH-2A/SH2A-FPU CPU		Example of Other CPU

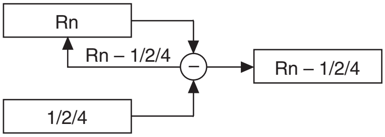
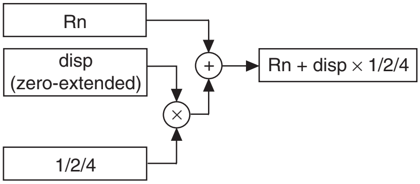
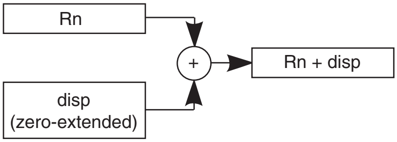
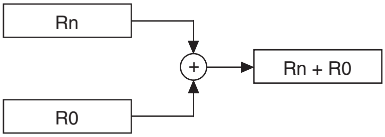
16-bit displacement	`MOV.W`
	`MOV.W`
	
	`.DATA.W`
		`@(disp,PC),R0`
		`@(R0,R1),R2`
		`...................`
		`H'1234`
			`MOV.W   @(H'1234,R1),R2`

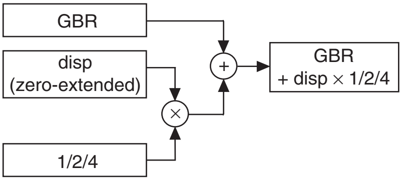
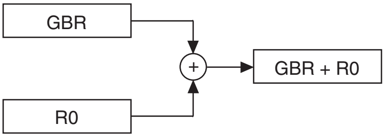
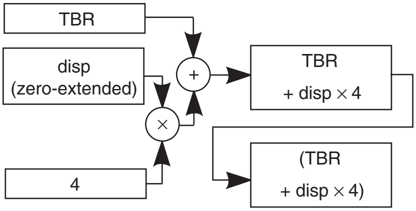
2.4.2 Addressing Modes

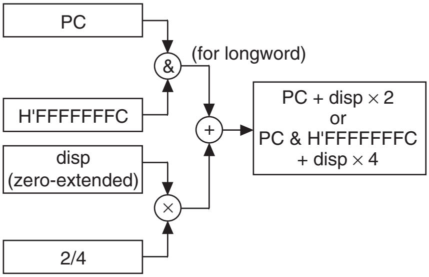
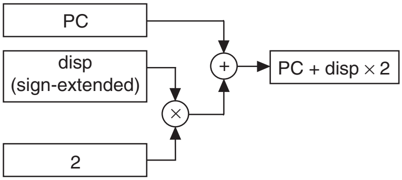
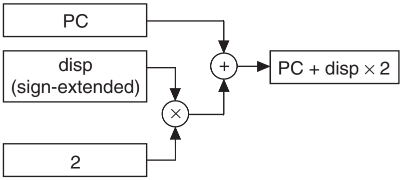
The addressing modes and effective address calculation methods are listed below.

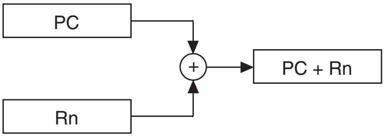
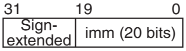
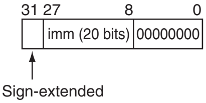
Table 2.9 Addressing Modes and Effective Addresses

Addressing Mode	Instruction Format	Effective Address Calculation	Equation
Register direct	`Rn`	The effective address is register Rn. (The operand is the contents of register Rn.)	—
Register indirect	`@Rn`	The effective address is the contents of register Rn. 	Rn
Register indirect with post-increment	`@Rn+`	The effective address is the contents of register Rn. A constant is added to the contents of Rn after the instruction is executed. 1 is added for a byte operation, 2 for a word operation, and 4 for a longword operation. 	Rn (After instruction execution) Byte: Rn + 1 → Rn Word: Rn + 2 → Rn Longword: Rn + 4 → Rn

Addressing Mode	Instruction Format	Effective Address Calculation	Equation
Register indirect with pre-decrement	`@-Rn`	The effective address is the value obtained by subtracting a constant from Rn. 1 is subtracted for a byte operation, 2 for a word operation, and 4 for a longword operation.  	Byte: Rn – 1 → Rn Word: Rn – 2 → Rn Longword: Rn – 4 → Rn (Instruction is executed with Rn after this calculation)
Register indirect with displacement	`@(disp:4, Rn)`	The effective address is the sum of Rn and a 4-bit displacement (disp). The value of disp is zero-extended, and remains unchanged for a byte operation, is doubled for a word operation, and is quadrupled for a longword operation.  	Byte: Rn + disp Word: Rn + disp × 2 Longword: Rn + disp × 4
Register indirect with displacement	`@(disp:12,Rn)`	The effective address is the sum of Rn and a 12-bit displacement (disp). The value of disp is zero-extended.  	Byte: Rn + disp Word: Rn + disp Longword: Rn + disp
Indexed register indirect	`@(R0,Rn)`	The effective address is the sum of Rn and R0.  	Rn + R0

Addressing Mode	Instruction Format	Effective Address Calculation	Equation
GBR indirect with displacement	`@(disp:8, GBR)`	The effective address is the sum of GBR value and an 8-bit displacement (disp). The value of disp is zero-extended, and remains unchanged for a byte operation, is doubled for a word operation, and is quadrupled for a longword operation.  GBR disp (zero-extended) 1/2/4 + × GBR + disp × 1/2/4	Byte: GBR + disp Word: GBR + disp × 2 Longword: GBR + disp × 4
Indexed GBR indirect	`@(R0, GBR)`	The effective address is the sum of GBR value and R0. 	GBR + R0
TBR duplicate indirect with displacement	`@@ (disp:8, TBR)`	The effective address is the sum of TBR value and an 8-bit displacement (disp). The value of disp is zero-extended, and is multiplied by 4.  TBR disp (zero-extended) 4 + × TBR + disp × 4 (TBR + disp × 4)	Contents of address (TBR + disp × 4)

Addressing Mode	Instruction Format	Effective Address Calculation	Equation
PC indirect with displacement	`@(disp:8, PC)`	The effective address is the sum of PC value and an 8-bit displacement (disp). The value of disp is zero-extended, and is doubled for a word operation, and quadrupled for a longword operation. For a longword operation, the lowest two bits of the PC value are masked.  PC (for longword) & H'FFFFFFFC + PC + disp × 2 or PC & H'FFFFFFFC + disp × 4 disp (zero-extended) × 2/4	Word: PC + disp × 2 Longword: PC & H'FFFFFFFC + disp × 4
PC relative	`disp:8`	The effective address is the sum of PC value and the value that is obtained by doubling the sign-extended 8-bit displacement (disp).  PC disp (sign-extended) + PC + disp × 2 × 2	PC + disp × 2
	`disp:12`	The effective address is the sum of PC value and the value that is obtained by doubling the sign-extended 12-bit displacement (disp).  PC disp (sign-extended) + PC + disp × 2 × 2	PC + disp × 2

Addressing Mode	Instruction Format	Effective Address Calculation	Equation
PC relative	`Rn`	The effective address is the sum of PC value and Rn.  	PC + Rn
Immediate	`#imm:20`	The 20-bit immediate data (imm) for the MOVI20 instruction is sign-extended.  	—
		The 20-bit immediate data (imm) for the MOVI20S instruction is shifted by eight bits to the left, the upper bits are sign-extended, and the lower bits are padded with zero.  31 27 8 0 imm (20 bits) 00000000 Sign-extended	—
	`#imm:8`	The 8-bit immediate data (imm) for the TST, AND, OR, and XOR instructions is zero-extended.	—
	`#imm:8`	The 8-bit immediate data (imm) for the MOV, ADD, and CMP/EQ instructions is sign-extended.	—
	`#imm:8`	The 8-bit immediate data (imm) for the TRAPA instruction is zero-extended and then quadrupled.	—
	`#imm:3`	The 3-bit immediate data (imm) for the BAND, BOR, BXOR, BST, BLD, BSET, and BCLR instructions indicates the target bit location.	—

2.4.3 Instruction Format

The instruction formats and the meaning of source and destination operands are described below. The meaning of the operand depends on the instruction code. The symbols used are as follows:

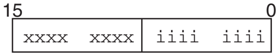
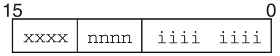
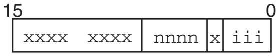
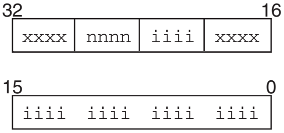
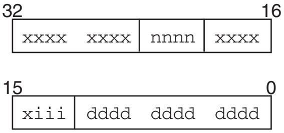
- `xxxx`: Instruction code
- `mmmm`: Source register
- `nnnn`: Destination register
- `iiii`: Immediate data
- `dddd`: Displacement

Table 2.10 Instruction Formats

Instruction Formats	Source Operand	Destination Operand	Example
0 format 15 0 xxxx xxxx xxxx xxxx	—	—	`NOP`
n format 15 0 xxxx nnnn xxxx xxxx	—	`nnnn`: Register direct	`MOVT Rn`
	Control register or system register	`nnnn`: Register direct	`STS MACH,Rn`
	R0 (Register direct)	`nnnn`: Register direct	`DIVU R0,Rn`
	Control register or system register	`nnnn`: Register indirect with pre-decrement	`STC.L SR,@-Rn`
	`mmmm`: Register direct	R15 (Register indirect with pre-decrement)	`MOVMU.L Rm,@-R15`
	R15 (Register indirect with post-increment)	`nnnn`: Register direct	`MOVMU.L @R15+,Rn`
	R0 (Register direct)	`nnnn`: (Register indirect with post-increment)	`MOV.L R0,@Rn+`

Instruction Formats	Source Operand	Destination Operand	Example
m format 15 0 xxxx mmmm xxxx xxxx	mmmm: Register direct	Control register or system register	`LDC Rm,SR`
	mmmm: Register indirect with post-increment	Control register or system register	`LDC.L @Rm+,SR`
	mmmm: Register indirect	—	`JMP @Rm`
	mmmm: Register indirect with pre-decrement	R0 (Register direct)	`MOV.L @-Rm,R0`
	mmmm: PC relative using Rm	—	`BRAF Rm`
nm format 15 0 xxxx nnnn mmmm xxxx	mmmm: Register direct	nnnn: Register direct	`ADD Rm,Rn`
	mmmm: Register direct	nnnn: Register indirect	`MOV.L Rm,@Rn`
	mmmm: Register indirect with post-increment (multiply-and-accumulate) nnnn*: Register indirect with post-increment (multiply-and-accumulate)	MACH, MACL	`MAC.W @Rm+,@Rn+`
	mmmm: Register indirect with post-increment	nnnn: Register direct	`MOV.L @Rm+,Rn`
	mmmm: Register direct	nnnn: Register indirect with pre-decrement	`MOV.L Rm,@-Rn`
	mmmm: Register direct	nnnn: Indexed register indirect	`MOV.L Rm,@(R0,Rn)`
md format 15 0 xxxx xxxx mmmm dddd	mmmmdddd: Register indirect with displacement	R0 (Register direct)	`MOV.B @(disp,Rm),R0`

Instruction Formats	Source Operand	Destination Operand	Example
nd4 format 15 0 xxxx xxxx nnnn dddd	R0 (Register direct)	nnnndddd: Register indirect with displacement	MOV.B R0,@(disp,Rn)
nmd format 15 0 xxxx nnnn mmmm dddd	mmmm: Register direct	nnnndddd: Register indirect with displacement	MOV.L Rm,@(disp,Rn)
	mmmmdddd: Register indirect with displacement	nnnn: Register direct	MOV.L @(disp,Rm),Rn
nmd12 format 32 16 xxxx nnnn mmmm xxxx 15 0 xxxx dddd dddd dddd	mmmm: Register direct	nnnndddd: Register indirect with displacement	MOV.L Rm,@(disp12,Rn)
	mmmmdddd: Register indirect with displacement	nnnn: Register direct	MOV.L @(disp12,Rm),Rn
d format 15 0 xxxx xxxx dddd dddd	dddddddd: GBR indirect with displacement	R0 (Register direct)	MOV.L @(disp,GBR),R0
	R0 (Register direct)	dddddddd: GBR indirect with displacement	MOV.L R0,@(disp,GBR)
	dddddddd: PC relative with displacement	R0 (Register direct)	MOVA @(disp,PC),R0
	dddddddd: TBR duplicate indirect with displacement	—	JSR/N @@(disp8,TBR)
	dddddddd: PC relative	—	BF label
d12 format 15 0 xxxx dddd dddd dddd	dddddddddddd: PC relative	—	BRA label (label = disp + PC)
nd8 format 15 0 xxxx nnnn dddd dddd	dddddddd: PC relative with displacement	nnnn: Register direct	MOV.L @(disp,PC),Rn

Instruction Formats	Source Operand	Destination Operand	Example
i format 	iiiiiiii: Immediate	Indexed GBR indirect	AND.B #imm,@(R0,GBR)
	iiiiiiii: Immediate	R0 (Register direct)	AND #imm,R0
	iiiiiiii: Immediate	—	TRAPA #imm
ni format 	iiiiiiii: Immediate	nnnn: Register direct	ADD #imm,Rn
ni3 format 	nnnn: Register direct iii: Immediate	—	BLD #imm3,Rn
	—	nnnn: Register direct iii: Immediate	BST #imm3,Rn
ni20 format  32 16 xxxx nnnn iiii xxxx 15 0 iiii iiii iiii iiii	iiiiiiiiiiii iiiiiiii: Immediate	nnnn: Register direct	MOVI20 #imm20, Rn
nid format  32 16 xxxx xxxx nnnn xxxx 15 0 xiii dddd dddd dddd	nnnndddddddd dddd: Register indirect with displacement iii: Immediate	—	BLD.B #imm3,@(disp12,Rn)
	—	nnnndddddddddddd: Register indirect with displacement iii: Immediate	BST.B #imm3,@(disp12,Rn)

Note: * In multiply-and-accumulate instructions, nnnn is the source register.

2.5 Instruction Set

2.5.1 Instruction Set by Classification

Table 2.11 lists the instructions according to their classification.

Table 2.11 Classification of Instructions

Classification	Types	Operation Code	Function	No. of Instructions
Data transfer	13	MOV	Data transfer Immediate data transfer Peripheral module data transfer Structure data transfer Reverse stack transfer	62
		MOVA	Effective address transfer	
		MOVI20	20-bit immediate data transfer	
		MOVI20S	20-bit immediate data transfer 8-bit left-shit	
		MOVML	R0−Rn register save/restore	
		MOVMU	Rn−R14 and PR register save/restore	
		MOVRT	T bit inversion and transfer to Rn	
		MOVT	T bit transfer	
		MOVU	Unsigned data transfer	
		NOTT	T bit inversion	
		PREF	Prefetch to operand cache	
		SWAP	Swap of upper and lower bytes	
		XTRCT	Extraction of the middle of registers connected	

Classification	Types	Operation Code	Function	No. of Instructions
Arithmetic operations	26	ADD	Binary addition	40
		ADDC	Binary addition with carry	
		ADDV	Binary addition with overflow check	
		CMP/cond	Comparison	
		CLIPS	Signed saturation value comparison	
		CLIPU	Unsigned saturation value comparison	
		DIVS	Signed division (32 ÷ 32)	
		DIVU	Unsigned division (32 ÷ 32)	
		DIV1	One-step division	
		DIV0S	Initialization of signed one-step division	
		DIV0U	Initialization of unsigned one-step division	
		DMULS	Signed double-precision multiplication	
		DMULU	Unsigned double-precision multiplication	
		DT	Decrement and test	
		EXTS	Sign extension	
		EXTU	Zero extension	
		MAC	Multiply-and-accumulate, double-precision multiply-and-accumulate operation	
		MUL	Double-precision multiply operation	
		MULR	Signed multiplication with result storage in Rn	
		MULS	Signed multiplication	
		MULU	Unsigned multiplication	
		NEG	Negation	
		NEGC	Negation with borrow	
		SUB	Binary subtraction	
		SUBC	Binary subtraction with borrow	
		SUBV	Binary subtraction with underflow	

Classification	Types	Operation Code	Function	No. of Instructions
Logic operations	6	AND	Logical AND	14
		NOT	Bit inversion	
		OR	Logical OR	
		TAS	Memory test and bit set	
		TST	Logical AND and T bit set	
		XOR	Exclusive OR	
Shift	12	ROTL	One-bit left rotation	16
		ROTR	One-bit right rotation	
		ROTCL	One-bit left rotation with T bit	
		ROTCR	One-bit right rotation with T bit	
		SHAD	Dynamic arithmetic shift	
		SHAL	One-bit arithmetic left shift	
		SHAR	One-bit arithmetic right shift	
		SHLD	Dynamic logical shift	
		SHLL	One-bit logical left shift	
		SHLLn	n-bit logical left shift	
		SHLR	One-bit logical right shift	
		SHLRn	n-bit logical right shift	

Classification	Types	Operation Code	Function	No. of Instructions
Branch	10	BF	Conditional branch, conditional delayed branch (branch when T = 0)	15
		BT	Conditional branch, conditional delayed branch (branch when T = 1)	
		BRA	Unconditional delayed branch	
		BRAF	Unconditional delayed branch	
		BSR	Delayed branch to subroutine procedure	
		BSRF	Delayed branch to subroutine procedure	
		JMP	Unconditional delayed branch	
		JSR	Branch to subroutine procedure Delayed branch to subroutine procedure	
		RTS	Return from subroutine procedure Delayed return from subroutine procedure	
		RTV/N	Return from subroutine procedure with Rm → R0 transfer	
System control	14	CLRT	T bit clear	36
		CLRMAC	MAC register clear	
		LDBANK	Register restoration from specified register bank entry	
		LDC	Load to control register	
		LDS	Load to system register	
		NOP	No operation	
		RESBANK	Register restoration from register bank	
		RTE	Return from exception handling	
		SETT	T bit set	
		SLEEP	Transition to power-down mode	
		STBANK	Register save to specified register bank entry	
		STC	Store control register data	
		STS	Store system register data	
		TRAPA	Trap exception handling	

Classification	Types	Operation Code	Function	No. of Instructions
Floating-point instructions	19	FABS	Floating-point absolute value	48
		FADD	Floating-point addition	
		FCMP	Floating-point comparison	
		FCNVDS	Conversion from double-precision to single-precision	
		FCNVSD	Conversion from single-precision to double -precision	
		FDIV	Floating-point division	
		FLDI0	Floating-point load immediate 0	
		FLDI1	Floating-point load immediate 1	
		FLDS	Floating-point load into system register FPUL	
		FLOAT	Conversion from integer to floating-point	
		FMAC	Floating-point multiply and accumulate operation	
		FMOV	Floating-point data transfer	
		FMUL	Floating-point multiplication	
		FNEG	Floating-point sign inversion	
		FSCHG	SZ bit inversion	
		FSQRT	Floating-point square root	
		FSTS	Floating-point store from system register FPUL	
		FSUB	Floating-point subtraction	
		FTRC	Floating-point conversion with rounding to integer	

Classification	Types	Operation Code	Function	No. of Instructions
FPU-related CPU instructions	2	LDS	Load into floating-point system register	8
		STS	Store from floating-point system register	
Bit manipulation	10	BAND	Bit AND	14
		BCLR	Bit clear	
		BLD	Bit load	
		BOR	Bit OR	
		BSET	Bit set	
		BST	Bit store	
		BXOR	Bit exclusive OR	
		BANDNOT	Bit NOT AND	
		BORNOT	Bit NOT OR	
		BLDNOT	Bit NOT load	
Total:	112			253

The table below shows the format of instruction codes, operation, and execution states. They are described by using this format according to their classification.

Instruction	Instruction Code	Operation	Execution Cycles	T Bit
Indicated by mnemonic.	Indicated in MSB ↔ LSB order.	Indicates summary of operation.	Value when no wait states are inserted.*1	Value of T bit after instruction is executed.
[Legend]	[Legend]	[Legend]		[Legend]
OP.Sz SRC, DEST OP: Operation code Sz: Size SRC: Source DEST: Destination Rm: Source register Rn: Destination register imm: Immediate data disp: Displacement*2	mmmm: Source register nnnn: Destination register 0000: R0 0001: R1 ········· 1111: R15 iiii: Immediate data dddd: Displacement	→, ←: Transfer direction (xx): Memory operand M/Q/T: Flag bits in SR &: Logical AND of each bit |: Logical OR of each bit ^: Exclusive logical OR of each bit ~: Logical NOT of each bit <<n: n-bit left shift >>n: n-bit right shift		—: No change

Notes: 1. Instruction execution cycles: The execution cycles shown in the table are minimums. In practice, the number of instruction execution states will be increased in cases such as the following:

a. When there is a conflict between an instruction fetch and a data access

b. When the destination register of a load instruction (memory → register) is the same as the register used by the next instruction.

2. Depending on the operand size, displacement is scaled by ×1, ×2, or ×4. For details, refer to the SH-2A, SH2A-FPU Software Manual.

2.5.2 Data Transfer Instructions

Table 2.12 Data Transfer Instructions

Instruction		Instruction Code	Operation	Execu-tion Cycles	T Bit	Compatibility		
						SH2E	SH4	SH-2A/ SH2A-FPU
MOV	#imm, Rn	1110nnnniiiiiiii	imm → sign extension → Rn	1	—	Yes	Yes	
MOV.W	@(disp, PC),Rn	1001nnnndddddddd	(disp × 2 + PC) → sign extension → Rn	1	—	Yes	Yes	
MOV.L	@(disp, PC),Rn	1101nnnndddddddd	(disp × 4 + PC) → Rn	1	—	Yes	Yes	
MOV	Rm, Rn	0110nnnnmmmm0011	Rm → Rn	1	—	Yes	Yes	
MOV.B	Rm, @Rn	0010nnnnmmmm0000	Rm → (Rn)	1	—	Yes	Yes	
MOV.W	Rm, @Rn	0010nnnnmmmm0001	Rm → (Rn)	1	—	Yes	Yes	
MOV.L	Rm, @Rn	0010nnnnmmmm0010	Rm → (Rn)	1	—	Yes	Yes	
MOV.B	@Rm, Rn	0110nnnnmmmm0000	(Rm) → sign extension → Rn	1	—	Yes	Yes	
MOV.W	@Rm, Rn	0110nnnnmmmm0001	(Rm) → sign extension → Rn	1	—	Yes	Yes	
MOV.L	@Rm, Rn	0110nnnnmmmm0010	(Rm) → Rn	1	—	Yes	Yes	
MOV.B	Rm, @-Rn	0010nnnnmmmm0100	Rn-1 → Rn, Rm → (Rn)	1	—	Yes	Yes	
MOV.W	Rm, @-Rn	0010nnnnmmmm0101	Rn-2 → Rn, Rm → (Rn)	1	—	Yes	Yes	
MOV.L	Rm, @-Rn	0010nnnnmmmm0110	Rn-4 → Rn, Rm → (Rn)	1	—	Yes	Yes	
MOV.B	@Rm+, Rn	0110nnnnmmmm0100	(Rm) → sign extension → Rn, Rm + 1 → Rm	1	—	Yes	Yes	
MOV.W	@Rm+, Rn	0110nnnnmmmm0101	(Rm) → sign extension → Rn, Rm + 2 → Rm	1	—	Yes	Yes	
MOV.L	@Rm+, Rn	0110nnnnmmmm0110	(Rm) → Rn, Rm + 4 → Rm	1	—	Yes	Yes	
MOV.B	R0, @(disp,Rn)	10000000nnnndddd	R0 → (disp + Rn)	1	—	Yes	Yes	
MOV.W	R0, @(disp,Rn)	10000001nnnndddd	R0 → (disp × 2 + Rn)	1	—	Yes	Yes	
MOV.L	Rm, @(disp,Rn)	0001nnnnmmmmdddd	Rm → (disp × 4 + Rn)	1	—	Yes	Yes	
MOV.B	@(disp, Rm),R0	10000100mmmmdddd	(disp + Rm) → sign extension → R0	1	—	Yes	Yes	

						Compatibility		
Instruction		Instruction Code	Operation	Execution Cycles	T Bit	SH2E	SH4	SH-2A/ SH2A-FPU
MOV.W	@(disp, Rm),R0	10000101mmmmdddd	(disp × 2 + Rm) → sign extension → R0	1	—	Yes	Yes	
MOV.L	@(disp, Rm),Rn	0101nnnnmmmmdddd	(disp × 4 + Rm) → Rn	1	—	Yes	Yes	
MOV.B	Rm,@(R0,Rn)	0000nnnnmmmm0100	Rm → (R0 + Rn)	1	—	Yes	Yes	
MOV.W	Rm,@(R0,Rn)	0000nnnnmmmm0101	Rm → (R0 + Rn)	1	—	Yes	Yes	
MOV.L	Rm,@(R0,Rn)	0000nnnnmmmm0110	Rm → (R0 + Rn)	1	—	Yes	Yes	
MOV.B	@(R0,Rm),Rn	0000nnnnmmmm1100	(R0 + Rm) → sign extension → Rn	1	—	Yes	Yes	
MOV.W	@(R0,Rm),Rn	0000nnnnmmmm1101	(R0 + Rm) → sign extension → Rn	1	—	Yes	Yes	
MOV.L	@(R0,Rm),Rn	0000nnnnmmmm1110	(R0 + Rm) → Rn	1	—	Yes	Yes	
MOV.B	R0,@(disp,GBR)	11000000dddddddd	R0 → (disp + GBR)	1	—	Yes	Yes	
MOV.W	R0,@(disp,GBR)	11000001dddddddd	R0 → (disp × 2 + GBR)	1	—	Yes	Yes	
MOV.L	R0,@(disp,GBR)	11000010dddddddd	R0 → (disp × 4 + GBR)	1	—	Yes	Yes	
MOV.B	@(disp,GBR),R0	11000100dddddddd	(disp + GBR) → sign extension → R0	1	—	Yes	Yes	
MOV.W	@(disp,GBR),R0	11000101dddddddd	(disp × 2 + GBR) → sign extension → R0	1	—	Yes	Yes	
MOV.L	@(disp,GBR),R0	11000110dddddddd	(disp × 4 + GBR) → R0	1	—	Yes	Yes	
MOV.B	R0,@Rn+	0100nnnn10001011	R0 → (Rn), Rn + 1 → Rn	1	—			Yes
MOV.W	R0,@Rn+	0100nnnn10011011	R0 → (Rn), Rn + 2 → Rn	1	—			Yes
MOV.L	R0,@Rn+	0100nnnn10101011	R0 → Rn), Rn + 4 → Rn	1	—			Yes
MOV.B	@-Rm,R0	0100mmmm11001011	Rm-1 → Rm, (Rm) → sign extension → R0	1	—			Yes
MOV.W	@-Rm,R0	0100mmmm11011011	Rm-2 → Rm, (Rm) → sign extension → R0	1	—			Yes
MOV.L	@-Rm,R0	0100mmmm11101011	Rm-4 → Rm, (Rm) → R0	1	—			Yes
MOV.B	Rm,@(disp12,Rn)	0011nnnnmmmm0001 0000dddddddddddd	Rm → (disp + Rn)	1	—			Yes
MOV.W	Rm,@(disp12,Rn)	0011nnnnmmmm0001 0001dddddddddddd	Rm → (disp × 2 + Rn)	1	—			Yes

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility		
						SH2E	SH4	SH-2A/ SH2A-FPU
MOV.L	Rm,@(disp12,Rn)	0011nnnnmmmm0001 0010dddddddddddd	Rm → (disp × 4 + Rn)	1	—			Yes
MOV.B	@(disp12, Rm), Rn	0011nnnnmmmm0001 0100dddddddddddd	(disp + Rm) → sign extension → Rn	1	—			Yes
MOV.W	@(disp12, Rm), Rn	0011nnnnmmmm0001 0101dddddddddddd	(disp × 2 + Rm) → sign extension → Rn	1	—			Yes
MOV.L	@(disp12, Rm), Rn	0011nnnnmmmm0001 0110dddddddddddd	(disp × 4 + Rm) → Rn	1	—			Yes
MOVA	@(disp,PC),R0	11000111dddddddd	disp × 4 + PC → R0	1	—	Yes	Yes	
MOVI20	#imm20, Rn	0000nnnniiii0000 iiiiiiiiiiiiiiii	imm → sign extension → Rn	1	—			Yes
MOVI20S	#imm20, Rn	0000nnnniiii0001 iiiiiiiiiiiiiiii	imm << 8 → sign extension → Rn	1	—			Yes
MOVML.L	Rm, @-R15	0100mmmm11110001	R15-4 → R15, Rm → (R15) R15-4 → R15, Rm-1 → (R15) : R15-4 → R15, R0 → (R15) Note: When Rm = R15, read Rm as PR	1 to 16	—			Yes
MOVML.L	@R15+, Rn	0100nnnn11110101	(R15) → R0, R15 + 4 → R15 (R15) → R1, R15 + 4 → R15 : (R15) → Rn Note: When Rn = R15, read Rm as PR	1 to 16	—			Yes

					Compatibility		
Instruction	Instruction Code	Operation	Execution Cycles	T Bit	SH2E	SH4	SH-2A/ SH2A-FPU
MOVMU.L Rm, @-R15	0100mmmm11110000	R15-4 → R15, PR → (R15) R15-4 → R15, R14 → (R15) : R15-4 → R15, Rm → (R15) Note: When Rm = R15, read Rm as PR	1 to 16	—			Yes
MOVMU.L @R15+, Rn	0100nnnn11110100	(R15) → Rn, R15 + 4 → R15 (R15) → Rn + 1, R15 + 4 → R15 : (R15) → R14, R15 + 4 → R15 (R15) → PR Note: When Rn = R15, read Rm as PR	1 to 16	—			Yes
MOVRT Rn	0000nnnn00111001	~T → Rn	1	—			Yes
MOVT Rn	0000nnnn00101001	T → Rn	1	—	Yes	Yes	
MOVU.B @(disp12, Rm), Rn	0011nnnnmmmm0001 1000dddddddddddd	(disp + Rm) → zero extension → Rn	1	—			Yes
MOVU.W @(disp12, Rm), Rn	0011nnnnmmmm0001 1001dddddddddddd	(disp × 2 + Rm) → zero extension → Rn	1	—			Yes
NOTT	0000000001101000	~T → T	1	Operation result			Yes
PREF @Rn	0000nnnn10000011	(Rn) → operand cache	1	—		Yes	
SWAP.B Rm, Rn	0110nnnnmmmm1000	Rm → swap lower 2 bytes → Rn	1	—	Yes	Yes	
SWAP.W Rm, Rn	0110nnnnmmmm1001	Rm → swap upper and lower words → Rn	1	—	Yes	Yes	
XTRCT Rm, Rn	0010nnnnmmmm1101	Middle 32 bits of Rm:Rn → Rn	1	—	Yes	Yes	

2.5.3 Arithmetic Operation Instructions

Table 2.13 Arithmetic Operation Instructions

						Compatibility		
Instruction		**Instruction Code**	**Operation**	**Execution Cycles**	**T Bit**	**SH2E**	**SH4**	**SH-2A/ SH2A-FPU**
ADD	Rm, Rn	`0011nnnnmmmm1100`	Rn + Rm → Rn	1	—	Yes	Yes	
ADD	#imm, Rn	`0111nnnniiiiiiii`	Rn + imm → Rn	1	—	Yes	Yes	
ADDC	Rm, Rn	`0011nnnnmmmm1110`	Rn + Rm + T → Rn, carry → T	1	Carry	Yes	Yes	
ADDV	Rm, Rn	`0011nnnnmmmm1111`	Rn + Rm → Rn, overflow → T	1	Overflow	Yes	Yes	
CMP/EQ	#imm, R0	`10001000iiiiiiii`	When R0 = imm, 1 → T Otherwise, 0 → T	1	Comparison result	Yes	Yes	
CMP/EQ	Rm, Rn	`0011nnnnmmmm0000`	When Rn = Rm, 1 → T Otherwise, 0 → T	1	Comparison result	Yes	Yes	
CMP/HS	Rm,Rn	`0011nnnnmmmm0010`	When Rn ≥ Rm (unsigned), 1 → T Otherwise, 0 → T	1	Comparison result	Yes	Yes	
CMP/GE	Rm, Rn	`0011nnnnmmmm0011`	When Rn ≥ Rm (signed), 1 → T Otherwise, 0 → T	1	Comparison result	Yes	Yes	
CMP/HI	Rm, Rn	`0011nnnnmmmm0110`	When Rn > Rm (unsigned), 1 → T Otherwise, 0 → T	1	Comparison result	Yes	Yes	
CMP/GT	Rm,Rn	`0011nnnnmmmm0111`	When Rn > Rm (signed), 1 → T Otherwise, 0 → T	1	Comparison result	Yes	Yes	
CMP/PL	Rn	`0100nnnn00010101`	When Rn > 0, 1 → T Otherwise, 0 → T	1	Comparison result	Yes	Yes	
CMP/PZ	Rn	`0100nnnn00010001`	When Rn ≥ 0, 1 → T Otherwise, 0 → T	1	Comparison result	Yes	Yes	

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility		
						SH2E	SH4	SH-2A/ SH2A-FPU
CMP/STR	Rm, Rn	0010nnnnmmmm1100	When any bytes are equal, 1 → T Otherwise, 0 → T	1	Com-parison result	Yes	Yes	
CLIPS.B	Rn	0100nnnn10010001	When Rn > (H'0000007F), (H'0000007F) → Rn, 1 → CS when Rn < (H'FFFFFF80), (H'FFFFFF80) → Rn, 1 → CS	1	—			Yes
CLIPS.W	Rn	0100nnnn10010101	When Rn > (H'00007FFF), (H'00007FFF) → Rn, 1 → CS When Rn < (H'FFFF8000), (H'FFFF8000) → Rn, 1 → CS	1	—			Yes
CLIPU.B	Rn	0100nnnn10000001	When Rn > (H'000000FF), (H'000000FF) → Rn, 1 → CS	1	—			Yes
CLIPU.W	Rn	0100nnnn10000101	When Rn > (H'0000FFFF), (H'0000FFFF) → Rn, 1 → CS	1	—			Yes
DIV1	Rm, Rn	0011nnnnmmmm0100	1-step division (Rn ÷ Rm)	1	Calcu-lation result	Yes	Yes	
DIV0S	Rm, Rn	0010nnnnmmmm0111	MSB of Rn → Q, MSB of Rm → M, M ^ Q → T	1	Calcu-lation result	Yes	Yes	
DIV0U		0000000000011001	0 → M/Q/T	1	0	Yes	Yes	
DIVS	R0, Rn	0100nnnn10010100	Signed operation of Rn ÷ R0 → Rn 32 ÷ 32 → 32 bits	36	—			Yes

					Compatibility		
Instruction	Instruction Code	Operation	Execution Cycles	T Bit	SH2E	SH4	SH-2A/ SH2A-FPU
DIVU R0, Rn	0100nnnn10000100	Unsigned operation of Rn ÷ R0 → Rn 32 ÷ 32 → 32 bits	34	—			Yes
DMULS.L Rm, Rn	0011nnnnmmmm1101	Signed operation of Rn × Rm → MACH, MACL 32 × 32 → 64 bits	2	—	Yes	Yes	
DMULU.L Rm, Rn	0011nnnnmmmm0101	Unsigned operation of Rn × Rm → MACH, MACL 32 × 32 → 64 bits	2	—	Yes	Yes	
DT Rn	0100nnnn00010000	Rn – 1 → Rn When Rn is 0, 1 → T When Rn is not 0, 0 → T	1	Comparison result	Yes	Yes	
EXTS.B Rm, Rn	0110nnnnmmmm1110	Byte in Rm is sign-extended → Rn	1	—	Yes	Yes	
EXTS.W Rm, Rn	0110nnnnmmmm1111	Word in Rm is sign-extended → Rn	1	—	Yes	Yes	
EXTU.B Rm, Rn	0110nnnnmmmm1100	Byte in Rm is zero-extended → Rn	1	—	Yes	Yes	
EXTU.W Rm, Rn	0110nnnnmmmm1101	Word in Rm is zero-extended → Rn	1	—	Yes	Yes	
MAC.L @Rm+, @Rn+	0000nnnnmmmm1111	Signed operation of (Rn) × (Rm) + MAC → MAC 32 × 32 + 64 → 64 bits	4	—	Yes	Yes	
MAC.W @Rm+, @Rn+	0100nnnnmmmm1111	Signed operation of (Rn) × (Rm) + MAC → MAC 16 × 16 + 64 → 64 bits	3	—	Yes	Yes	
MUL.L Rm, Rn	0000nnnnmmmm0111	Rn × Rm → MACL 32 × 32 → 32 bits	2	—	Yes	Yes	
MULR R0, Rn	0100nnnn10000000	R0 × Rn → Rn 32 × 32 → 32 bits	2				Yes
MULS.W Rm, Rn	0010nnnnmmmm1111	Signed operation of Rn × Rm → MACL 16 × 16 → 32 bits	1	—	Yes	Yes	

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility		
						SH2E	SH4	SH-2A/ SH2A-FPU
MULU.W	Rm, Rn	0010nnnnmmmm1110	Unsigned operation of Rn × Rm → MACL 16 × 16 → 32 bits	1	—	Yes	Yes	
NEG	Rm, Rn	0110nnnnmmmm1011	0-Rm → Rn	1	—	Yes	Yes	
NEGC	Rm, Rn	0110nnnnmmmm1010	0-Rm-T → Rn, borrow → T	1	Borrow	Yes	Yes	
SUB	Rm, Rn	0011nnnnmmmm1000	Rn-Rm → Rn	1	—	Yes	Yes	
SUBC	Rm, Rn	0011nnnnmmmm1010	Rn-Rm-T → Rn, borrow → T	1	Borrow	Yes	Yes	
SUBV	Rm, Rn	0011nnnnmmmm1011	Rn-Rm → Rn, underflow → T	1	Overflow	Yes	Yes	

2.5.4 Logic Operation Instructions

Table 2.14 Logic Operation Instructions

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility		
						SH2E	SH4	SH-2A/ SH2A-FPU
AND	Rm, Rn	0010nnnnmmmm1001	Rn & Rm → Rn	1	—	Yes	Yes	
AND	#imm, R0	11001001iiiiiiii	R0 & imm → R0	1	—	Yes	Yes	
AND.B	#imm, @(R0, GBR)	11001101iiiiiiii	(R0 + GBR) & imm → (R0 + GBR)	3	—	Yes	Yes	
NOT	Rm, Rn	0110nnnnmmmm0111	~Rm → Rn	1	—	Yes	Yes	
OR	Rm, Rn	0010nnnnmmmm1011	Rn | Rm → Rn	1	—	Yes	Yes	
OR	#imm, R0	11001011iiiiiiii	R0 | imm → R0	1	—	Yes	Yes	
OR.B	#imm, @(R0, GBR)	11001111iiiiiiii	(R0 + GBR) | imm → (R0 + GBR)	3	—	Yes	Yes	
TAS.B	@Rn	0100nnnn00011011	When (Rn) is 0, 1 → T Otherwise, 0 → T, 1 → MSB of(Rn)	3	Test result	Yes	Yes	

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility		
						SH2E	SH4	SH-2A/ SH2A-FPU
TST	Rm, Rn	0010nnnnmmmm1000	Rn & Rm When the result is 0, 1 → T Otherwise, 0 → T	1	Test result	Yes	Yes	
TST	#imm, R0	11001000iiiiiiii	R0 & imm When the result is 0, 1 → T Otherwise, 0 → T	1	Test result	Yes	Yes	
TST.B	#imm, @(R0, GBR)	11001100iiiiiiii	(R0 + GBR) & imm When the result is 0, 1 → T Otherwise, 0 → T	3	Test result	Yes	Yes	
XOR	Rm, Rn	0010nnnnmmmm1010	Rn ^ Rm → Rn	1	—	Yes	Yes	
XOR	#imm, R0	11001010iiiiiiii	R0 ^ imm → R0	1	—	Yes	Yes	
XOR.B	#imm, @(R0, GBR)	11001110iiiiiiii	(R0 + GBR) ^ imm → (R0 + GBR)	3	—	Yes	Yes	

2.5.5 Shift Instructions

Table 2.15 Shift Instructions

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility		
						SH2E	SH4	SH-2A/ SH2A-FPU
ROTL	Rn	0100nnnn00000100	T ← Rn ← MSB	1	MSB	Yes	Yes	
ROTR	Rn	0100nnnn00000101	LSB → Rn → T	1	LSB	Yes	Yes	
ROTCL	Rn	0100nnnn00100100	T ← Rn ← T	1	MSB	Yes	Yes	
ROTCR	Rn	0100nnnn00100101	T → Rn → T	1	LSB	Yes	Yes	
SHAD	Rm, Rn	0100nnnnmmmm1100	When Rm ≥ 0, Rn << Rm → Rn When Rm < 0, Rn >> |Rm| → [MSB → Rn]	1	—		Yes	

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility		
						SH2E	SH4	SH-2A/ SH2A-FPU
SHAL	Rn	`0100nnnn00100000`	T ← Rn ← 0	1	MSB	Yes	Yes	
SHAR	Rn	`0100nnnn00100001`	MSB → Rn → T	1	LSB	Yes	Yes	
SHLD	Rm, Rn	`0100nnnnmmmm1101`	When Rm ≥ 0, Rn << Rm → Rn When Rm < 0, Rn >> |Rm| → [0 → Rn]	1	—		Yes	
SHLL	Rn	`0100nnnn00000000`	T ← Rn ← 0	1	MSB	Yes	Yes	
SHLR	Rn	`0100nnnn00000001`	0 → Rn → T	1	LSB	Yes	Yes	
SHLL2	Rn	`0100nnnn00001000`	Rn << 2 → Rn	1	—	Yes	Yes	
SHLR2	Rn	`0100nnnn00001001`	Rn >> 2 → Rn	1	—	Yes	Yes	
SHLL8	Rn	`0100nnnn00011000`	Rn << 8 → Rn	1	—	Yes	Yes	
SHLR8	Rn	`0100nnnn00011001`	Rn >> 8 → Rn	1	—	Yes	Yes	
SHLL16	Rn	`0100nnnn00101000`	Rn << 16 → Rn	1	—	Yes	Yes	
SHLR16	Rn	`0100nnnn00101001`	Rn >> 16 → Rn	1	—	Yes	Yes	

2.5.6 Branch Instructions

Table 2.16 Branch Instructions

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility		
						SH2E	SH4	SH-2A/ SH2A-FPU
BF	label	`10001011dddddddd`	When T = 0, disp × 2 + PC → PC, When T = 1, nop	3/1*	—	Yes	Yes	
BF/S	label	`10001111dddddddd`	Delayed branch When T = 0, disp × 2 + PC → PC, When T = 1, nop	2/1*	—	Yes	Yes	
BT	label	`10001001dddddddd`	When T = 1, disp × 2 + PC → PC, When T = 0, nop	3/1*	—	Yes	Yes	

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility		
						SH2E	SH4	SH-2A/ SH2A-FPU
BT/S	label	10001101dddddddd	Delayed branch When T = 1, disp × 2 + PC → PC, When T = 0, nop	2/1*	—	Yes	Yes	
BRA	label	1010dddddddddddd	Delayed branch, disp × 2 + PC → PC	2	—	Yes	Yes	
BRAF	Rm	0000mmmm00100011	Delayed branch, Rm + PC → PC	2	—	Yes	Yes	
BSR	label	1011dddddddddddd	Delayed branch, PC → PR, disp × 2 + PC → PC	2	—	Yes	Yes	
BSRF	Rm	0000mmmm00000011	Delayed branch, PC → PR, Rm + PC → PC	2	—	Yes	Yes	
JMP	@Rm	0100mmmm00101011	Delayed branch, Rm → PC	2	—	Yes	Yes	
JSR	@Rm	0100mmmm00001011	Delayed branch, PC → PR, Rm → PC	2	—	Yes	Yes	
JSR/N	@Rm	0100mmmm01001011	PC-2 → PR, Rm → PC	3	—			Yes
JSR/N	@@(disp8, TBR)	10000011dddddddd	PC-2 → PR, (disp × 4 + TBR) → PC	5	—			Yes
RTS		0000000000001011	Delayed branch, PR → PC	2	—	Yes	Yes	
RTS/N		0000000001101011	PR → PC	3	—			Yes
RTV/N	Rm	0000mmmm01111011	Rm → R0, PR → PC	3	—			Yes

Note: * One cycle when the program does not branch.

2.5.7 System Control Instructions

Table 2.17 System Control Instructions

					Compatibility		
Instruction	**Instruction Code**	**Operation**	**Execution Cycles**	**T Bit**	**SH2E**	**SH4**	**SH-2A/ SH2A-FPU**
CLRT	`0000000000001000`	0 → T	1	0	Yes	Yes	
CLRMAC	`0000000000101000`	0 → MACH,MACL	1	—	Yes	Yes	
LDBANK @Rm,R0	`0100mmmm11100101`	(Specified register bank entry) → R0	6	—			Yes
LDC Rm,SR	`0100mmmm00001110`	Rm → SR	3	LSB	Yes	Yes	
LDC Rm,TBR	`0100mmmm01001010`	Rm → TBR	1	—			Yes
LDC Rm,GBR	`0100mmmm00011110`	Rm → GBR	1	—	Yes	Yes	
LDC Rm,VBR	`0100mmmm00101110`	Rm → VBR	1	—	Yes	Yes	
LDC.L @Rm+,SR	`0100mmmm00000111`	(Rm) → SR, Rm + 4 → Rm	5	LSB	Yes	Yes	
LDC.L @Rm+,GBR	`0100mmmm00010111`	(Rm) → GBR, Rm + 4 → Rm	1	—	Yes	Yes	
LDC.L @Rm+,VBR	`0100mmmm00100111`	(Rm) → VBR, Rm + 4 → Rm	1	—	Yes	Yes	
LDS Rm,MACH	`0100mmmm00001010`	Rm → MACH	1	—	Yes	Yes	
LDS Rm,MACL	`0100mmmm00011010`	Rm → MACL	1	—	Yes	Yes	
LDS Rm,PR	`0100mmmm00101010`	Rm → PR	1	—	Yes	Yes	
LDS.L @Rm+,MACH	`0100mmmm00000110`	(Rm) → MACH, Rm + 4 → Rm	1	—	Yes	Yes	
LDS.L @Rm+,MACL	`0100mmmm00010110`	(Rm) → MACL, Rm + 4 → Rm	1	—	Yes	Yes	
LDS.L @Rm+,PR	`0100mmmm00100110`	(Rm) → PR, Rm + 4 → Rm	1	—	Yes	Yes	
NOP	`0000000000001001`	No operation	1	—	Yes	Yes	
RESBANK	`0000000001011011`	Bank → R0 to R14, GBR, MACH, MACL, PR	9*	—			Yes
RTE	`0000000000101011`	Delayed branch, stack area → PC/SR	6	—	Yes	Yes	

Instruction		Instruction Code	Operation	Execu-tion Cycles	T Bit	Compatibility SH2E	SH4	SH-2A/ SH2A-FPU
SETT		`0000000000011000`	1 → T	1	1	Yes	Yes	
SLEEP		`0000000000011011`	Sleep	5	—	Yes	Yes	
STBANK	R0,@Rn	`0100nnnn11100001`	R0 → (specified register bank entry)	7	—			Yes
STC	SR,Rn	`0000nnnn00000010`	SR → Rn	2	—	Yes	Yes	
STC	TBR,Rn	`0000nnnn01001010`	TBR → Rn	1	—			Yes
STC	GBR,Rn	`0000nnnn00010010`	GBR → Rn	1	—	Yes	Yes	
STC	VBR,Rn	`0000nnnn00100010`	VBR → Rn	1	—	Yes	Yes	
STC.L	SR,@-Rn	`0100nnnn00000011`	Rn-4 → Rn, SR → (Rn)	2	—	Yes	Yes	
STC.L	GBR,@-Rn	`0100nnnn00010011`	Rn-4 → Rn, GBR → (Rn)	1	—	Yes	Yes	
STC.L	VBR,@-Rn	`0100nnnn00100011`	Rn-4 → Rn, VBR → (Rn)	1	—	Yes	Yes	
STS	MACH,Rn	`0000nnnn00001010`	MACH → Rn	1	—	Yes	Yes	
STS	MACL,Rn	`0000nnnn00011010`	MACL → Rn	1	—	Yes	Yes	
STS	PR,Rn	`0000nnnn00101010`	PR → Rn	1	—	Yes	Yes	
STS.L	MACH,@-Rn	`0100nnnn00000010`	Rn-4 → Rn, MACH → (Rn)	1	—	Yes	Yes	
STS.L	MACL,@-Rn	`0100nnnn00010010`	Rn-4 → Rn, MACL → (Rn)	1	—	Yes	Yes	
STS.L	PR,@-Rn	`0100nnnn00100010`	Rn-4 → Rn, PR → (Rn)	1	—	Yes	Yes	
TRAPA	#imm	`11000011iiiiiiii`	PC/SR → stack area, (imm × 4 + VBR) → PC	5	—	Yes	Yes	

Notes: Instruction execution cycles: The execution cycles shown in the table are minimums. In practice, the number of instruction execution states in cases such as the following:

a. When there is a conflict between an instruction fetch and a data access

b. When the destination register of a load instruction (memory → register) is the same as the register used by the next instruction.

* In the event of bank overflow, the number of cycles is 19.

2.5.8 Floating-Point Operation Instructions

Table 2.18 Floating-Point Operation Instructions

						Compatibility		
Instruction		Instruction Code	Operation	Execution Cycles	T Bit	SH2E	SH4	SH-2A/ SH2A-FPU
FABS	FRn	1111nnnn01011101	|FRn| → FRn	1	—	Yes	Yes	
FABS	DRn	1111nnn001011101	|DRn| → DRn	1	—		Yes	
FADD	FRm, FRn	1111nnnnmmmm0000	FRn + FRm → FRn	1	—	Yes	Yes	
FADD	DRm, DRn	1111nnn0mmm00000	DRn + DRm → DRn	6	—		Yes	
FCMP/EQ	FRm, FRn	1111nnnnmmmm0100	(FRn = FRm)? 1:0 → T	1	Comparison result	Yes	Yes	
FCMP/EQ	DRm, DRn	1111nnn0mmm00100	(DRn = DRm)? 1:0 → T	2	Comparison result		Yes	
FCMP/GT	FRm, FRn	1111nnnnmmmm0101	(FRn > FRm)? 1:0 → T	1	Compa -rison result	Yes	Yes	
FCMP/GT	DRm, DRn	1111nnn0mmm00101	(DRn > DRm)? 1:0 → T	2	Comparison result		Yes	
FCNVDS	DRm, FPUL	1111mmm010111101	(float) DRm → FPUL	2	—		Yes	
FCNVSD	FPUL, DRn	1111nnn010101101	(double) FPUL → DRn	2	—		Yes	
FDIV	FRm, FRn	1111nnnnmmmm0011	FRn/FRm → FRn	10	—	Yes	Yes	
FDIV	DRm, DRn	1111nnn0mmm00011	DRn/DRm → DRn	23	—		Yes	
FLDI0	FRn	1111nnnn10001101	0 × 00000000 → FRn	1	—	Yes	Yes	
FLDI1	FRn	1111nnnn10011101	0 × 3F800000 → FRn	1	—	Yes	Yes	
FLDS	FRm, FPUL	1111mmmm00011101	FRm → FPUL	1	—	Yes	Yes	
FLOAT	FPUL,FRn	1111nnnn00101101	(float)FPUL → FRn	1	—	Yes	Yes	
FLOAT	FPUL,DRn	1111nnn000101101	(double)FPUL → DRn	2	—		Yes	
FMAC	FR0,FRm,FRn	1111nnnnmmmm1110	FR0 × FRm+FRn → FRn	1	—	Yes	Yes	
FMOV	FRm, FRn	1111nnnnmmmm1100	FRm → FRn	1	—	Yes	Yes	
FMOV	DRm, DRn	1111nnn0mmm01100	DRm → DRn	2	—		Yes	

					Compatibility		
Instruction	Instruction Code	Operation	Execu-tion Cycles	T Bit	SH2E	SH4	SH-2A/ SH2A-FPU
FMOV.S @(R0, Rm), FRn	1111nnnnmmmm0110	(R0 + Rm) → FRn	1	—	Yes	Yes	
FMOV.D @(R0, Rm), DRn	1111nnn0mmmm0110	(R0 + Rm) → DRn	2	—		Yes	
FMOV.S @Rm+, FRn	1111nnnnmmmm1001	(Rm) → FRn, Rm+=4	1	—	Yes	Yes	
FMOV.D @Rm+, DRn	1111nnn0mmmm1001	(Rm) → DRn, Rm += 8	2	—		Yes	
FMOV.S @Rm, FRn	1111nnnnmmmm1000	(Rm) → FRn	1	—	Yes	Yes	
FMOV.D @Rm, DRn	1111nnn0mmmm1000	(Rm) → DRn	2	—		Yes	
FMOV.S @(disp12,Rm),FRn	0011nnnnmmmm0001 0111dddddddddddd	(disp × 4 + Rm) → FRn	1	—			Yes
FMOV.D @(disp12,Rm),DRn	0011nnn0mmmm0001 0111dddddddddddd	(disp × 8 + Rm) → DRn	2	—			Yes
FMOV.S FRm, @(R0,Rn)	1111nnnnmmmm0111	FRm → (R0 + Rn)	1	—	Yes	Yes	
FMOV.D DRm, @(R0,Rn)	1111nnnnmmm00111	DRm → (R0 + Rn)	2	—		Yes	
FMOV.S FRm, @-Rn	1111nnnnmmmm1011	Rn-=4, FRm → (Rn)	1	—	Yes	Yes	
FMOV.D DRm, @-Rn	1111nnnnmmm01011	Rn-=8, DRm → (Rn)	2	—		Yes	
FMOV.S FRm, @Rn	1111nnnnmmmm1010	FRm → (Rn)	1	—	Yes	Yes	
FMOV.D DRm, @Rn	1111nnnnmmm01010	DRm → (Rn)	2	—		Yes	
FMOV.S FRm, @(disp12,Rn)	0011nnnnmmmm0001 0011dddddddddddd	FRm → (disp × 4 + Rn)	1	—			Yes
FMOV.D DRm, @(disp12,Rn)	0011nnnnmmm00001 0011dddddddddddd	DRm → (disp × 8 + Rn)	2	—			Yes
FMUL FRm, FRn	1111nnnnmmmm0010	FRn × FRm → FRn	1	—	Yes	Yes	
FMUL DRm, DRn	1111nnn0mmm00010	DRn × DRm → DRn	6	—		Yes	
FNEG FRn	1111nnnn01001101	-FRn → FRn	1	—	Yes	Yes	
FNEG DRn	1111nnn001001101	-DRn → DRn	1	—		Yes	
FSCHG	1111001111111101	FPSCR.SZ=~FPSCR.SZ	1	—		Yes	
FSQRT FRn	1111nnnn01101101	√FRn → FRn	9	—		Yes	
FSQRT DRn	1111nnn001101101	√DRn → DRn	22	—		Yes	
FSTS FPUL,FRn	1111nnnn00001101	FPUL → FRn	1	—	Yes	Yes	
FSUB FRm, FRn	1111nnnnmmmm0001	FRn-FRm → FRn	1	—	Yes	Yes	

Instruction		Instruction Code	Operation	Execu-tion Cycles	T Bit	Compatibility		
						SH2E	SH4	SH-2A/ SH2A-FPU
FSUB	DRm, DRn	1111nnn0mmm00001	DRn-DRm → DRn	6	—		Yes	
FTRC	FRm, FPUL	1111mmmm00111101	(long)FRm → FPUL	1	—	Yes	Yes	
FTRC	DRm, FPUL	1111mmm000111101	(long)DRm → FPUL	2	—		Yes	

2.5.9 FPU-Related CPU Instructions

Table 2.19 FPU-Related CPU Instructions

Instruction		Instruction Code	Operation	Execu-tion Cycles	T Bit	Compatibility		
						SH2E	SH4	SH-2A/ SH2A-FPU
LDS	Rm,FPSCR	0100mmmm01101010	Rm → FPSCR	1	—	Yes	Yes	
LDS	Rm,FPUL	0100mmmm01011010	Rm → FPUL	1	—	Yes	Yes	
LDS.L	@Rm+, FPSCR	0100mmmm01100110	(Rm) → FPSCR, Rm+=4	1	—	Yes	Yes	
LDS.L	@Rm+, FPUL	0100mmmm01010110	(Rm) → FPUL, Rm+=4	1	—	Yes	Yes	
STS	FPSCR, Rn	0000nnnn01101010	FPSCR → Rn	1	—	Yes	Yes	
STS	FPUL,Rn	0000nnnn01011010	FPUL → Rn	1	—	Yes	Yes	
STS.L	FPSCR,@-Rn	0100nnnn01100010	Rn-=4, FPCSR → (Rn)	1	—	Yes	Yes	
STS.L	FPUL,@-Rn	0100nnnn01010010	Rn-=4, FPUL → (Rn)	1	—	Yes	Yes	

2.5.10 Bit Manipulation Instructions

Table 2.20 Bit Manipulation Instructions

Instruction		Instruction Code	Operation	Execu-tion Cycles	T Bit	Compatibility		
						SH2E	SH4	SH-2A/ SH2A-FPU
BAND.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 0100dddddddddddd	(imm of (disp + Rn)) & T → T	3	Ope-ration result			Yes

					Compatibility		
Instruction	Instruction Code	Operation	Execution Cycles	T Bit	SH2E	SH4	SH-2A/ SH2A-FPU
BANDNOT.B #imm3,@(disp12,Rn)	0011nnnn0iii1001 1100dddddddddddd	~(imm of (disp + Rn)) & T → T	3	Operation result			Yes
BCLR.B #imm3,@(disp12,Rn)	0011nnnn0iii1001 0000dddddddddddd	0 → (imm of (disp + Rn))	3	—			Yes
BCLR #imm3,Rn	10000110nnnn0iii	0 → imm of Rn	1	—			Yes
BLD.B #imm3,@(disp12,Rn)	0011nnnn0iii1001 0011dddddddddddd	(imm of (disp + Rn)) →	3	Operation result			Yes
BLD #imm3,Rn	10000111nnnn1iii	imm of Rn → T	1	Operation result			Yes
BLDNOT.B #imm3,@(disp12,Rn)	0011nnnn0iii1001 1011dddddddddddd	~(imm of (disp + Rn)) → T	3	Operation result			Yes
BOR.B #imm3,@(disp12,Rn)	0011nnnn0iii1001 0101dddddddddddd	(imm of (disp + Rn)) | T → T	3	Operation result			Yes
BORNOT.B #imm3,@(disp12,Rn)	0011nnnn0iii1001 1101dddddddddddd	~(imm of (disp + Rn)) | T → T	3	Operation result			Yes
BSET.B #imm3,@(disp12,Rn)	0011nnnn0iii1001 0001dddddddddddd	1 → (imm of (disp + Rn))	3	—			Yes
BSET #imm3,Rn	10000110nnnn1iii	1 → imm of Rn	1	—			Yes
BST.B #imm3,@(disp12,Rn)	0011nnnn0iii1001 0010dddddddddddd	T → (imm of (disp + Rn))	3	—			Yes
BST #imm3,Rn	10000111nnnn0iii	T → imm of Rn	1	—			Yes
BXOR.B #imm3,@(disp12,Rn)	0011nnnn0iii1001 0110dddddddddddd	(imm of (disp + Rn)) ^ T → T	3	Operation result			Yes

2.6 Processing States

The CPU has four processing states: reset, exception handling, program execution, and power-down. Figure 2.8 shows the transitions between the states.

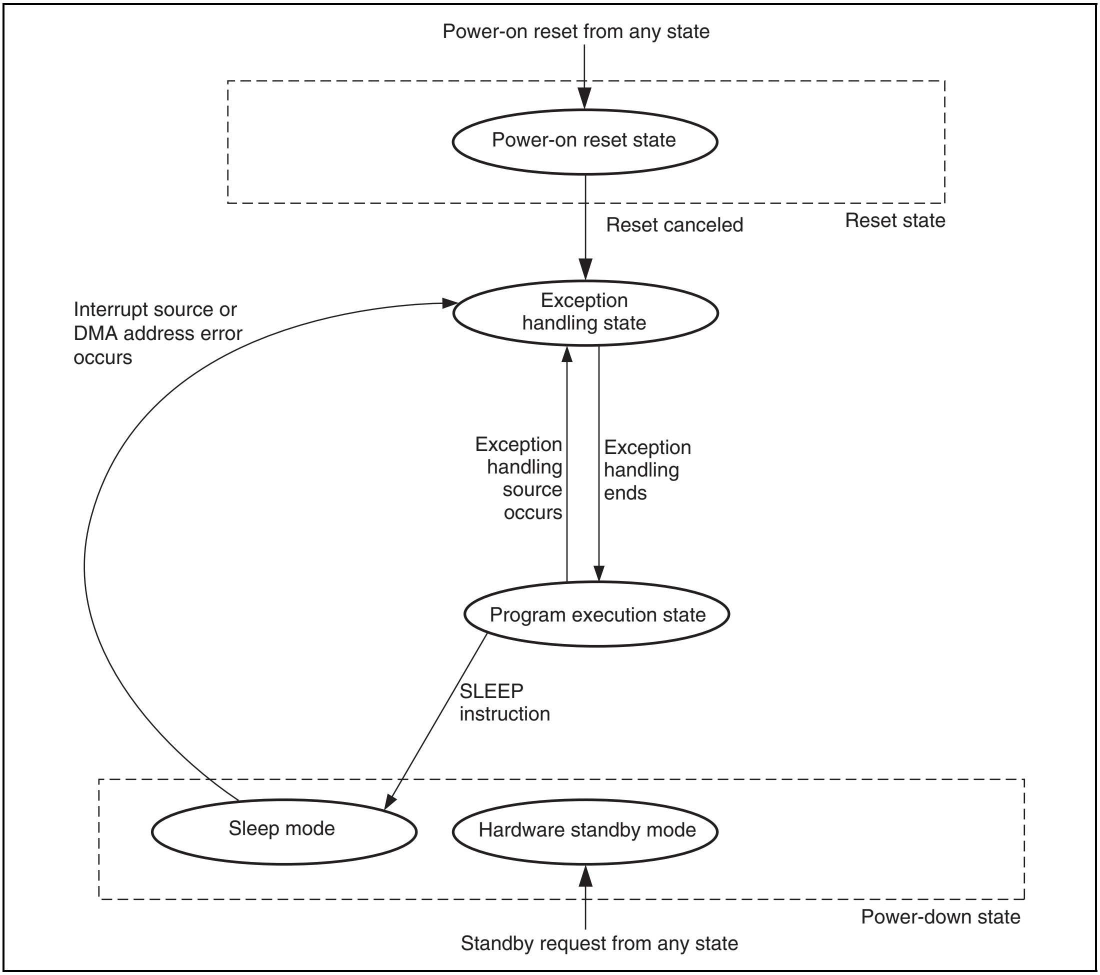


Figure 2.8 Transitions between Processing States

(1) Reset State

In this state, the CPU is reset by a power-on reset.

(2) Exception Handling State

The exception handling state is a transient state that occurs when exception handling sources such as resets or interrupts alter the CPU's processing state flow.

For a reset, the initial values of the program counter (PC) (execution start address) and stack pointer (SP) are fetched from the exception handling vector table and stored; the CPU then branches to the execution start address and execution of the program begins.

For an interrupt, the stack pointer (SP) is accessed and the program counter (PC) and status register (SR) are saved to the stack area. The exception service routine start address is fetched from the exception handling vector table; the CPU then branches to that address and the program starts executing, thereby entering the program execution state.

(3) Program Execution State

In the program execution state, the CPU sequentially executes the program.

(4) Power-Down State

In the power-down state, the CPU stops operating to conserve power. Sleep mode is entered by executing a SLEEP instruction. If hardware standby input is received, the CPU enters the hardware standby mode.

Section 3 Operating Modes

This LSI has a JTAG interface, so the $\overline{\text{TRST}}$ signal must be held low for a specified time when power is supplied. This is the case whether or not the JTAG interface is used, even in normal operating mode.

3.1 Types of Operating Modes and Selection

This LSI has six types of operating modes. The operating mode is determined by the setting of pins MD4 to MD0 and FWE. The setting should not be changed during LSI operation. However, the shift of mode from the MCU extended mode with the on-chip ROM enabled or the MCU single-chip mode to the user program mode is supported even when the FWE pin is operating. These pins should be set as shown in table 3.2.

The voltage of the PVcc1 power should range within the values shown in table 3.1.

Table 3.1 Selection of Operating Modes

Mode No.	Mode Name	On-Chip ROM	External Bus	Bus Width (Area 0)	Write-Access to ROM	PVcc1 Voltage
Mode 0	MCU extended mode	Disabled	Enabled	8 bits	—	3.3 V ±0.3 V
Mode 1		Disabled	Enabled	16 bits	—	3.3 V ±0.3 V
Mode 2		Enabled	Enabled	Set by CS0BCR*	Disabled	3.3 V ±0.3 V
Mode 3	MCU single-chip mode	Enabled	Disabled	—	Disabled	5.0 V ±0.5 V
Mode 4	Boot mode	Enabled	Enabled	Set by CS0BCR*	Enabled	3.3 V ±0.3 V
Mode 5		Enabled	Disabled	—	Enabled	5.0 V ±0.5 V
Mode 6	User program mode	Enabled	Enabled	Set by CS0BCR*	Enabled	3.3 V ±0.3 V
Mode 7		Enabled	Disabled	—	Enabled	5.0 V ±0.5 V
Mode 8	User boot mode	Enabled	Enabled	Set by CS0BCR*	Enabled	3.3 V ±0.3 V
Mode 9		Enabled	Disabled	—	Enabled	5.0 V ±0.5 V

Note: * The CSn space bus control register (CS0BCR) is a register of the bus state controller (BSC). The bus width can be selected from 8 bits or 16 bits.

Table 3.2 Operating Mode Pin Settings

		Pin Setting				
Mode No.	**Mode Name**	**MD4/MD3*1**	**MD2**	**MD1**	**MD0**	**FWE**
Mode 0	MCU extended mode	0	1	1	1	0*2
Mode 1			1	1	1	1*2
Mode 2			0	0	1	0
Mode 3	MCU single-chip mode		0	0	0	0
Mode 4	Boot mode		0	1	1	1
Mode 5			0	1	0	1
Mode 6	User program mode		0	0	1	1
Mode 7			0	0	0	1
Mode 8	User boot mode		1	0	1	1
Mode 9			1	0	0	1

Notes: 1. Pins MD4 and MD3 should always be driven to a level of 0.

2. The FWE pin functions as a mode setting pin that is used to select the external bus width in on-chip ROM disabled mode.

There are two modes as the MCU operating modes: the MCU extended mode and single-chip mode.

There are three modes as the flash memory programming modes: on-board programming modes (boot mode, user boot mode, and user program mode).

Independent from the above operating modes, this LSI features an ASE mode, which allows debugging operations through the external connection of an emulator. Setting the ASEMD pin to a level of 1 makes the LSI the ASE mode. If, for example, the MCU single-chip mode is selected through the settings of pins MD4 to MD0 and FWE, this LSI operates in MCU single chip mode of ASE mode.

If the LSI is set in ASE mode without connecting an emulator, the correct operation cannot be guaranteed. If this LSI is not connected to an emulator, the ASEMD pin must be set to a level of 0 so that this LSI operates in normal operating mode.

Unless otherwise noted, descriptions given in this manual assume that this LSI operates in normal operating mode.

Section 4 Clock Pulse Generator (CPG)

4.1 Overview

The clock pulse generator (CPG) supplies clock pulses to both the inside of this LSI and external devices. The CPG consists of an oscillation circuit and a PLL multiplier. There are two methods of generating a clock with the CPG: by connecting a crystal resonator or by inputting an external clock.

The oscillation circuit oscillates at the same frequency as the input clock.

Three types of clock signals are internally supplied: the internal clock (ϕ), peripheral clock (Pϕ), and FlexRay clock (Fϕ). The internal clock (ϕ) signal is supplied to the modules such as the CPU, FPU, on-chip RAM, and ROM cache. The frequency of this clock is selected from four, six, eight, or ten times the frequency of the clock signal input on the EXTAL pin. The multiplication ratio can only be changed through the settings of pins MD_CLK1 and MD_CLK0 and cannot be changed during LSI operation.

The peripheral clock signal (Pϕ) is mainly supplied to the on-chip peripheral modules. The frequency of this clock is selected from one or two times the frequency of the input clock from the EXTAL pin. The multiplication ratio can only be changed by setting MD_CLKP and cannot be changed during LSI operation. The CK pin outputs the peripheral clock signal (Pϕ).

The FlexRay clock (Fϕ) is supplied to the FlexRay module. The frequency of this clock is provided by frequency-multiplying (by four) the clock signal from the EXTAL pin.
The CPG is halted in hardware standby mode.

A block diagram of the CPG is shown in figure 4.1.

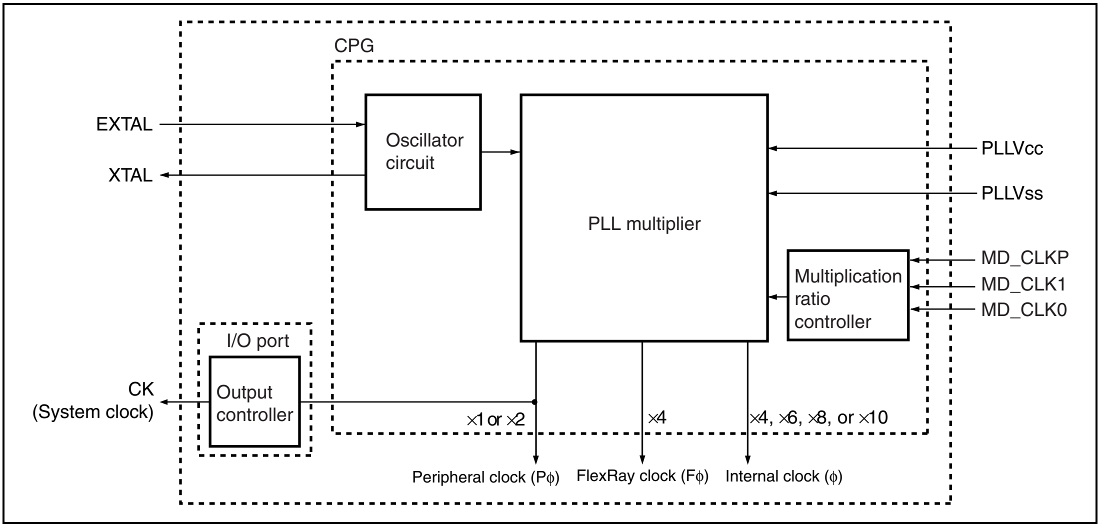


Figure 4.1 Block Diagram of CPG

4.2 Pin Configuration

Table 4.1 shows the pin configuration of the CPG.

Table 4.1 Pin Configuration

Pin Name	Symbol	I/O	Function
External clock	EXTAL	Input	Input pin for crystal resonator or external clock
Crystal	XTAL	Output	Input pin for crystal resonator
System clock	CK	Output	Output pin for system clock
Clock mode setting	MD_CLKP	Input	Input pin for setting peripheral clock frequency
	MD_CLK0, MD_CLK1	Input	Input pins for setting PLL multiplication ratio
PLL power supply	$PLLV_{CC}$	Input	Power supply pin for PLL multiplier
PLL ground	$PLLV_{SS}$	Input	Ground pin for PLL multiplier

4.3 Frequency Ranges and Clock Selection

Three types of clock signals, internal clock (ϕ), peripheral clock (Pϕ), and FlexRay clock (Fϕ) signals, are internally supplied.

This clock is supplied to the modules such as the CPU, FPU, on-chip RAM, and ROM cache. The frequency of the internal clock (ϕ) signal is selected from four, six, eight, or ten times the frequency of the clock signal input on the EXTAL pin according to the settings of pins MD_CLK1 and MD_CLK0.

The peripheral clock signal (Pϕ) is mainly supplied to the on-chip peripheral modules and is selected from one or two times the input frequency on the EXTAL pin according to the settings of the MD_CLKP pin. The CK pin outputs the peripheral clock signal (Pϕ).
The FlexRay clock (Fϕ) is supplied to the FlexRay module, and is obtained by frequency-multiplying the input clock by four.

The input frequency and operating frequency ranges for each pin setting are shown in table 4.2 and table 4.3.

The CK pin enables or disables the pin output through the setting of the CK control register (CKCR) of the I/O port. For details on CKCR, see section 24, I/O Ports.

Table 4.2 Frequency Range and Internal / FlexRay Clock selection

Pin Setting		Input Frequency Range (MHz)	PLL Multiplication Factor	Internal Clock Frequency Range (MHz)	FlexRay Frequency Range (MHz)
MD_CLK1	MD_CLK0				
0	0	16 to 20	× 4	64 to 80	64 to 80
	1		× 6	96 to 120	
1	0		× 10	160 to 200	
	1		× 8	128 to 160	

Note: The multiplication ratio and pin settings must not be changed during operation of this LSI.

Table 4.3 Frequency Range and Peripheral Clock selection

Pin Setting	Input Frequency Range (MHz)	PLL Multiplication Factor	Peripheral Clock Frequency Range (MHz)
MD_CLK0			
0	16 to 20	× 1	16 to 20
1	16 to 20	× 2	32 to 40

4.4 Clock Source

A crystal resonator or an external clock can be selected as the clock source.

4.4.1 Connecting Crystal Resonator

(1) Circuit Configuration

Figure 4.2 shows an example of connecting a crystal resonator. Use the damping resistor (Rd) shown in table 4.4. An AT-cut parallel-resonance type crystal resonator should be used. Load capacitors (CL1, CL2) must be connected as shown in figure 4.2.

The clock pulses generated by the crystal resonator and internal oscillation circuit are sent to the PLL multiplier, where the clock signals are multiplied to produce the selected frequency and supplied internally and externally.

The crystal resonator manufacturer should be consulted concerning the compatibility between the crystal resonator and this LSI.

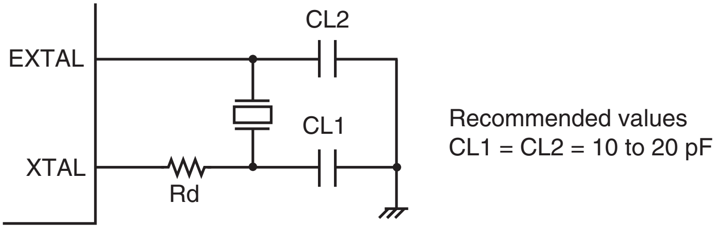


Figure 4.2 Connection Example of Crystal Resonator

Table 4.4 Damping Resistor Values (Recommended Values)

	Frequency (MHz)	
Parameter	**16**	**20**
Rd (Ω)	0	0

(2) Crystal Resonator

Figure 4.3 shows an equivalent circuit of the crystal resonator. Use a crystal resonator with the characteristics listed in table 4.5.

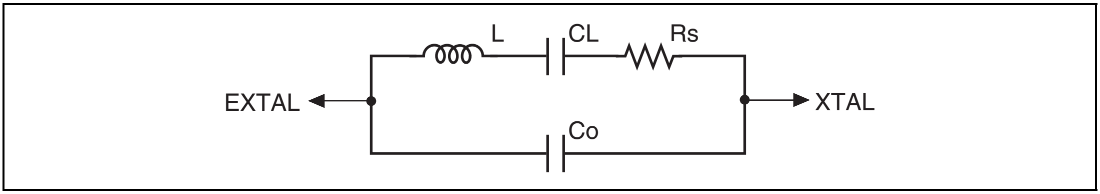


Figure 4.3 Crystal Resonator Equivalent Circuit

Table 4.5 Crystal Resonator Parameters (Recommended Values)

	Frequency (MHz)	
Parameter	**16**	**20**
Rs (Ω)	22	20
Co (pF)	1	1

4.4.2 External Clock Input

An example of external clock input connection is shown in figure 4.4.

Leave the XTAL pin open-circuit, but ensure that the parasitic capacitance on the XTAL pin is not greater than 1 pF.

Even when an external clock is input, secure the oscillation stabilization time when switching on or leaving the standby mode for PLL stabilization.

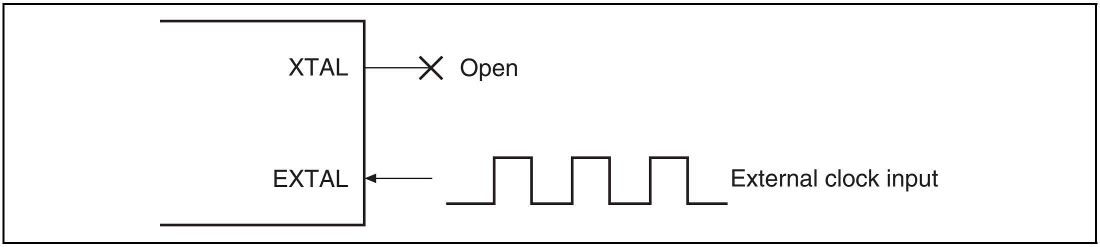


Figure 4.4 External Clock Input

4.5 Usage Notes

4.5.1 Note on Board Design

Place the crystal resonator and its load capacitors as close as possible to the XTAL and EXTAL pins.

To prevent induction from interfering with correct oscillation, do not allow any signal lines to cross the XTAL or EXTAL lines (figure 4.5).

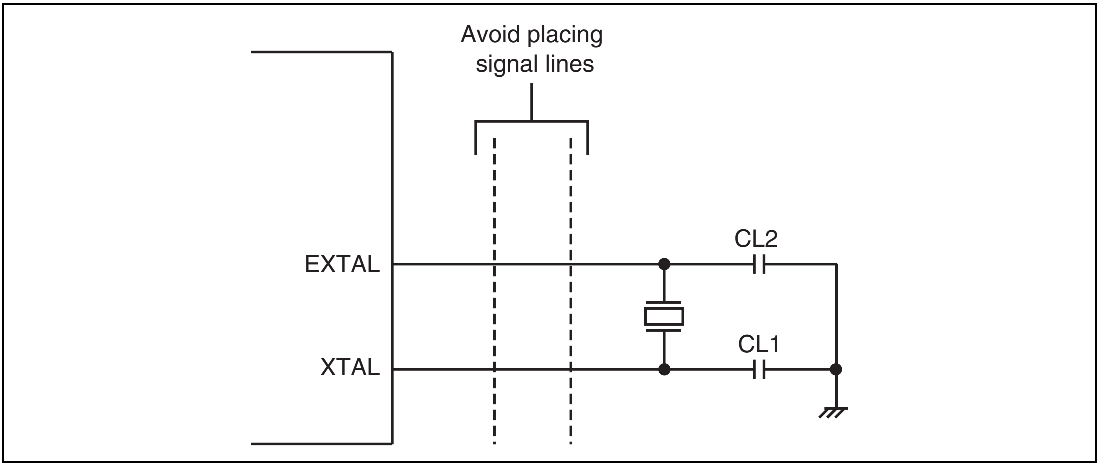


Figure 4.5 Note on Board Design

4.5.2 Note on Connecting Power Supply for PLL Oscillator

Separate $PLLV_{CC}$ and $PLLV_{SS}$ from the other V_{CC} and V_{SS} lines at the board power supply source, and be sure to insert bypass capacitors C_{PB} and C_B close to the pins.

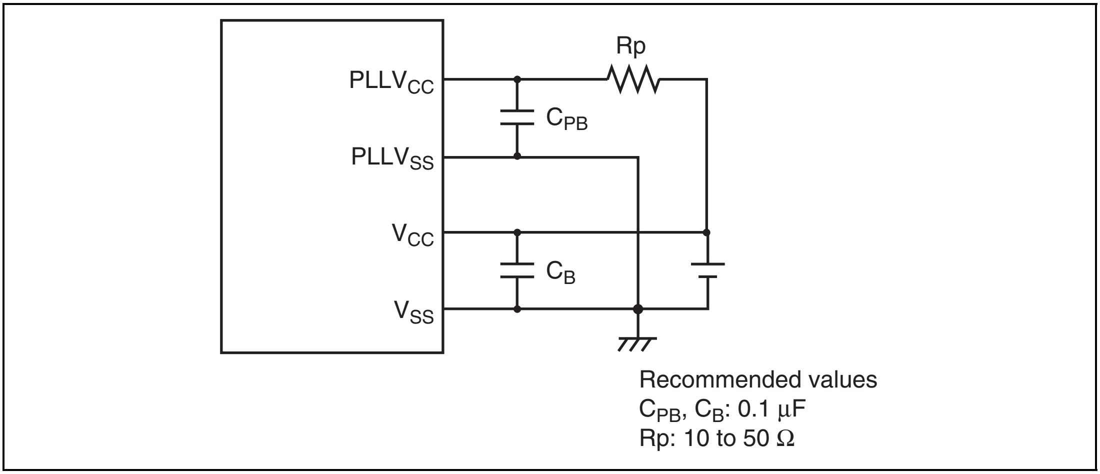


Figure 4.6 Note on Connecting Power Supply for PLL Oscillator

4.5.3 Note on Using the FlexRay

Since the frequency for FlexRay is 80 MHz, the frequency of the external clock must be 20 MHz if the FlexRay interface is to be used.

Section 5 Address Space

Table 5.1 shows the address space of the SH72567R in on-chip ROM enabled mode (exclusive of the single-chip mode); table 5.2 shows the address space of the SH72567R in on-chip ROM disabled mode; and table 5.3 shows the address space of the SH72567R in single-chip mode.

Access to the internal I/O register spaces can be made via the addresses shown in section 33, List of Registers. Access should not be made to any other addresses than those provided in the list and to reserved areas; otherwise, operation cannot be guaranteed.

Table 5.1 Address Space of the SH72567R in On-Chip ROM Enabled Mode (Exclusive of Single-Chip Mode)

Address	Type of address space	Size
H'0000 0000 to H'003F FFFF (H'0000 0000 to H'0000 7FFF)*[1]	On-chip ROM (read from user MAT) (on-chip ROM (read from user boot MAT))	4 Mbytes (32 Kbytes)
H'0040 0000 to H'0040 1FFF	Reserved area	
H'0040 2000 to H'0040 3FFF	FCU firmware area	8 Kbytes
H'0040 4000 to H'01FF FFFF	Reserved area	
H'0200 0000 to H'023F FFFF	External address space (CS0)	4 Mbytes
H'0240 0000 to H'03FF FFFF	CS0 shadow space	28 Mbytes
H'0400 0000 to H'043F FFFF	External address space (CS1)	4 Mbytes
H'0440 0000 to H'07FF FFFF	CS1 shadow space	60 Mbytes
H'0800 0000 to H'083F FFFF	External address space (CS2)	4 Mbytes
H'0840 0000 to H'0BFF FFFF	CS2 shadow space	60 Mbytes
H'0C00 0000 to H'0C3F FFFF	External address space (CS3)	4 Mbytes
H'0C40 0000 to H'0FFF FFFF	CS3 shadow space	60 Mbytes
H'1000 0000 to H'800F FFFF	Reserved area	
H'8010 0000 to H'8011 FFFF	EEPROM (read/write)*[2]	128 Kbytes
H'8012 0000 to H'807F FFFF	Reserved area	
H'8080 0000 to H'80BF FFFF (H'8080 0000 to H'8080 7FFF)*[1]	On-chip ROM (write to user MAT) (on-chip ROM (write to user boot MAT))	4 Mbytes (32 Kbytes)
H'80C0 0000 to H'80FF 7FFF	Reserved area	
H'80FF 8000 to H'80FF 9FFF	FCU RAM area	8 Kbytes

Address	Type of address space	Size
H'80FF A000 to H'FFF7 FFFF	Reserved area	
H'FFF8 0000 to H'FFFB FFFF	On-chip RAM	256 Kbytes
H'FFFC 0000 to H'FFFF FFFF	Internal I/O register	256 Kbytes (Max.)

Notes: 1. When the user boot MAT of the on-chip ROM is selected. See section 26, ROM.

2. The EEPROM area includes the lot trace information. See section 28, EEPROM.

Table 5.2 Address Space of the SH72567R in On-Chip ROM Disabled Mode

Address	Type of address space	Size
H'0000 0000 to H'003F FFFF	External address space (CS0)	4 Mbytes
H'0040 0000 to H'03FF FFFF	CS0 shadow space	60 Mbytes
H'0400 0000 to H'043F FFFF	External address space (CS1)	4 Mbytes
H'0440 0000 to H'07FF FFFF	CS1 shadow space	60 Mbytes
H'0800 0000 to H'083F FFFF	External address space (CS2)	4 Mbytes
H'0840 0000 to H'0BFF FFFF	CS2 shadow space	60 Mbytes
H'0C00 0000 to H'0C3F FFFF	External address space (CS3)	4 Mbytes
H'0C40 0000 to H'0FFF FFFF	CS3 shadow space	60 Mbytes
H'1000 0000 to H'FFF7 FFFF	Reserved area	
H'FFF8 0000 to H'FFFB FFFF	On-chip RAM	256 Kbytes
H'FFFC 0000 to H'FFFF FFFF	Internal I/O register	256 Kbytes (Max.)

Table 5.3 Address Space of the SH72567R in Single-Chip Mode

Address	Type of address space	Size
H'0000 0000 to H'003F FFFF (H'0000 0000 to H'0000 7FFF)*[1]	On-chip ROM (read from user MAT) (on-chip ROM (read from user boot MAT))	4.0 Mbytes (32 Kbytes)
H'0040 0000 to H'0040 1FFF	Reserved area	
H'0040 2000 to H'0040 3FFF	FCU firmware area	8 Kbytes
H'0040 4000 to H'800F FFFF	Reserved area	
H'8010 0000 to H'8011 FFFF	EEPROM (read/write)*[2]	128 Kbytes
H'8012 0000 to H'807F FFFF	Reserved area	
H'8080 0000 to H'80BF FFFF (H'8080 0000 to H'8080 7FFF)*[1]	On-chip ROM (write to user MAT) (on-chip ROM (write to user boot MAT))	4 Mbytes (32 Kbytes)
H'80C0 0000 to H'80FF 7FFF	Reserved area	
H'80FF 8000 to H'80FF 9FFF	FCU RAM area	8 Kbytes
H'80FF A000 to H'FFF7 FFFF	Reserved area	
H'FFF8 0000 to H'FFFB FFFF	On-chip RAM	256 Kbytes
H'FFFC 0000 to H'FFFF FFFF	Internal I/O register	256 Kbytes (Max.)

Notes: 1. When the user boot MAT of the on-chip ROM is selected. See section 26, ROM.

2. The EEPROM area includes the lot trace information. See section 28, EEPROM.

Section 6 Reset

When the $\overline{\text{RES}}$ pin is driven low, the LSI enters the power-on reset state. If the $\overline{\text{RES}}$ pin is driven high while the LSI is placed in power-on reset state, the power-on reset state is cancelled and the CPU starts power-on reset exception handling.

6.1 Reset Operation

When the $\overline{\text{RES}}$ pin is driven low by the low level pulse longer than or equal to the noise cancellation width (t_{RESNCW}), the reset request is accepted. When the reset request is accepted, all pin states are reset and each pin enters reset state. Each pin state during reset is summarized in appendix A, Pin States.

Internal circuits including the CPU are reset 3 to 4 Pϕ cycles after reset acceptance.

The $\overline{\text{RES}}$ pin should be kept at the low level during at least t_{RESW} (t_{cyc}). Later, if the $\overline{\text{RES}}$ pin is driven high while it is kept low, the reset state of internal circuits is cancelled after 3 to 4 Pϕ cycles and the CPU starts power-on reset exception handling.

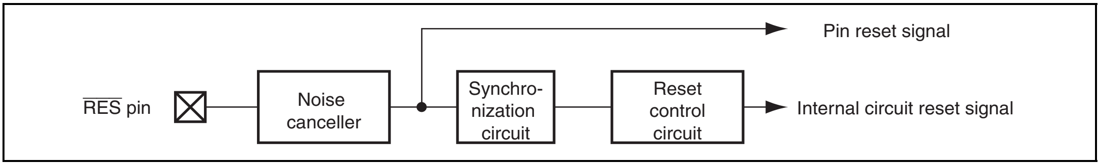


Figure 6.1 Reset Circuit

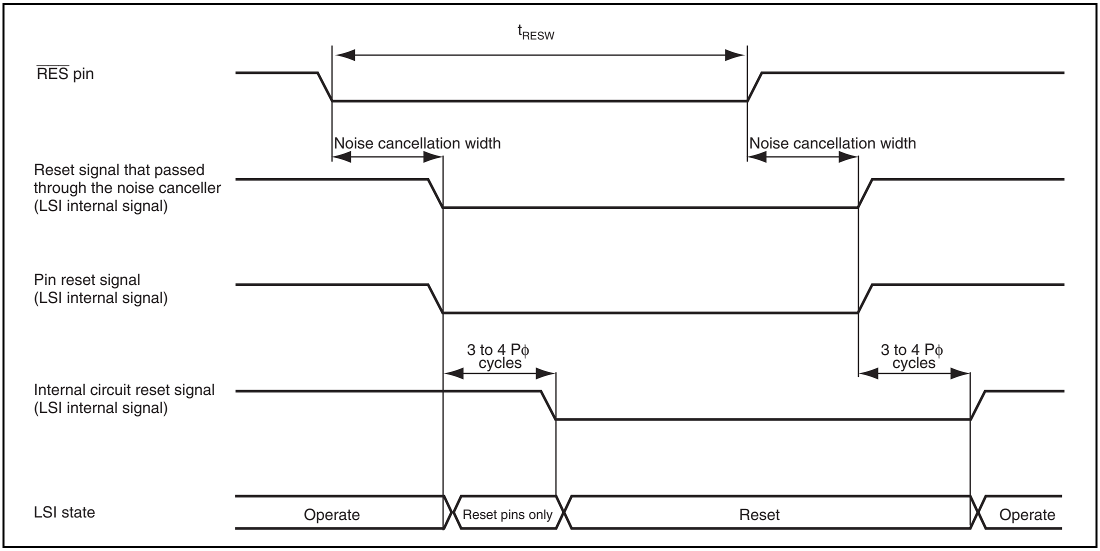

t_{RESW}
$\overline{RES}$ pin
Noise cancellation width
Noise cancellation width
Reset signal that passed through the noise canceller (LSI internal signal)
Pin reset signal (LSI internal signal)
3 to 4 Pφ cycles
3 to 4 Pφ cycles
Internal circuit reset signal (LSI internal signal)
LSI state
Operate
Reset pins only
Reset
Operate

Figure 6.2 Reset Sequence

6.1.1 Reset at Power-On

To reset at power-on, the $\overline{\text{RES}}$ pin should be kept at the low level for the duration of the internal PLL oscillation settling time after the power supply voltage falls within the specified range and the $\overline{\text{HSTBY}}$ pin is driven high. The internal PLL oscillation settling time is specified as t_{OSC1}. For details, see section 34, Electrical Characteristics.

6.1.2 Reset during Operation

To reset during operation, the $\overline{\text{RES}}$ pin should be kept at the low level for at least the reset pulse width. The reset pulse width is specified as t_{RESW}. For details, see section 34, Electrical Characteristics.

6.1.3 On-chip RAM Data Retention during Reset

When the $\overline{\text{RES}}$ pin is driven low causing a power-on reset while the internal bus master such as CPU accesses the RAM, the data at the accessed address may be lost. Data in addresses not accessed can be retained. To reliably retain the RAM data, the RAM should be disabled by the RAM enable register (RAMEN) before the $\overline{\text{RES}}$ pin is driven low. For details on the RAM enable register, see section 30, RAM.

6.2 Internal State after Reset Cancellation

Table 6.1 summarizes the internal state after reset cancellation. For details on initial state of each on-chip peripheral module registers, refer to each section.

Table 6.1 Initial Values of Register

Type		Register	Initial Value
CPU	General register	R0 to R14	Undefined
		R15 (SP)	SP value in the vector address table
	Control register	SR	I[3:0] = 1111 (H'F), BO = 0, CS = 0, reserved bits = 0, other bits = undefined
		GBR, TBR	Undefined
		VBR	H'0000 0000
	System register	MACH, MACL, PR	Undefined
		PC	PC value in the vector address table
	Floating point register	FPR0 to FPR15	Undefined
	Floating point system register	FPUL	Undefined
		FPSCR	H'00040001
RAM		—	Undefined after power-on

Section 7 Exception Handling

7.1 Overview

7.1.1 Types of Exception Handling and Priority

Exception handling is started by sources, such as a reset, address errors, a memory error, register bank errors, interrupts, and instructions. Table 7.1 shows their priorities. When several exception handling sources occur at once, they are processed according to the priority shown.

Table 7.1 Types of Exception Handling and Priority

Type	Exception Handling	Priority
Reset	Power-on reset	High
Address error	CPU address error	
	DMAC address error	
Instruction	FPU exception	
	Integer division exception (division by zero)	
	Integer division exception (overflow)	
Register bank error	Bank underflow	
	Bank overflow	
Interrupt	NMI	
	User break	
	IRQ	
	Memory error (RAM error/ROM error)	
	Software interrupt (SINT)	Low

Type	Exception Handling		Priority
Interrupt	On-chip peripheral modules	Direct memory access controller (DMAC)	High
		Compare match timer (CMT)	↑
		Watchdog timer (WDT)	
		Advanced timer unit III (ATU-III)	
		A/D converter (ADC)	
		Serial communications interface (SCI)	
		Renesas serial peripheral interface (RSPI)	
		Controller area network (RCAN-TL1)	
		Automotive direct memory access controller (A-DMAC)	
		FlexRay	
Instruction	Trap instruction (TRAPA instruction)		
	General illegal instructions (undefined code)		
	Slot illegal instructions (undefined code placed directly after a delayed branch instruction*[1] (including FPU instructions and FPU-related CPU instructions in FPU module standby status), instructions that rewrite the PC*[2], 32-bit instructions*[3], RESBANK instruction, DIVS instruction, and DIVU instruction)		↓ Low

Notes: 1. Delayed branch instructions: JMP, JSR, BRA, BSR, RTS, RTE, BF/S, BT/S, BSRF, and BRAF.

2. Instructions that rewrite the PC: JMP, JSR, BRA, BSR, RTS, RTE, BT, BF, TRAPA, BF/S, BT/S, BSRF, BRAF, JSR/N, and RTV/N.

3. 32-bit instructions: BAND.B, BANDNOT.B, BCLR.B, BLD.B, BLDNOT.B, BOR.B, BORNOT.B, BSET.B, BST.B, BXOR.B, FMOV.S@disp12, FMOV.D@disp12, MOV.B@disp12, MOV.W@disp12, MOV.L@disp12, MOVI20, MOVI20S, MOVU.B, and MOVU.W.

7.1.2 Exception Handling Operations

The exception handling sources are detected and exception handling starts according to the timing shown in table 7.2.

Table 7.2 Timing of Exception Source Detection and Start of Exception Handling

Exception	Source	Timing of Source Detection and Start of Handling
Reset	Power-on reset	Starts when the $\overline{\text{RES}}$ pin changes from low to high or when the WDT overflows.
Address error		Detected when instruction is decoded and starts when the previous executing instruction finishes executing.
Interrupts		
Register bank errors	Bank underflow	Starts upon attempted execution of a RESBANK instruction when saving has not been performed to register banks.
	Bank overflow	In the state where saving has been performed to all register bank areas, starts when acceptance of register bank overflow exception has been set by the interrupt controller (the BOVE bit in IBNR of the INTC is 1) and an interrupt that uses a register bank has occurred and been accepted by the CPU.
Instructions	Trap instruction	Starts from the execution of a TRAPA instruction.
	General illegal instructions	Starts from the decoding of undefined code anytime except immediately after a delayed branch instruction (delay slot) (including FPU instructions and FPU-related CPU instructions in FPU module standby status).
	Slot illegal instructions	Starts from the decoding of undefined code placed directly after a delayed branch instruction (delay slot) (including FPU instructions and FPU-related CPU instructions in FPU module standby status), of instructions that rewrite the PC, of 32-bit instructions, of the RESBANK instruction, of the DIVS instruction, or of the DIVU instruction.
	Integer division exceptions	Starts when detecting division-by-zero exception or overflow exception caused by division of the negative maximum value (H'80000000) by -1.
	FPU exceptions	Starts when detecting invalid operation exception defined by IEEE standard 754, division-by-zero exception, overflow, underflow, or inexact exception. Also starts when qNaN or $\pm\infty$ is input to the source for a floating point operation instruction when the QIS bit in FPSCR is set.

When exception handling starts, the CPU operates as follows.

(1) Exception Handling Triggered by Reset

The initial values of the program counter (PC) and stack pointer (SP) are fetched from the exception handling vector table (PC and SP are respectively the H'00000000 and H'00000004 addresses). See section 7.1.3, Exception Handling Vector Table, for the exception handling vector table. The vector base register (VBR) is then initialized to H'00000000, the interrupt mask level bits (I3 to I0) in the status register (SR) are initialized to H'F (B'1111), and the BO and CS bits are initialized. The BN bit in IBNR of the interrupt controller (INTC) is also initialized to 0. FPSCR is also initialized to H'00040001 by a power-on reset. The program starts running from the PC address fetched from the exception handling vector table.

(2) Exception Handling Triggered by Address Errors, Register Bank Errors, Interrupts, and Instructions

SR and PC are saved to the stack indicated by R15. In the case of interrupt exception handling other than NMI or user break with usage of the register banks enabled, general registers R0 to R14, control register GBR, system registers MACH, MACL, and PR, and the vector table address offset of the interrupt exception handling to be executed are saved to the register banks. In the case of exception handling due to an address error, register bank error, NMI interrupt, user break interrupt, or instruction, saving to a register bank is not performed. When saving is performed to all register banks, automatic saving to the stack is performed instead of register bank saving. In this case, an interrupt controller setting must have been made so that a register bank overflow exception is not accepted (the BOVE bit in IBNR of the INTC is 0). If a setting to accept a register bank overflow exception has been made (the BOVE bit in IBNR of the INTC is 1), the register bank overflow exception will be generated. In the case of interrupt exception handling, the interrupt priority level is written to the I3 to I0 bits in SR. In the case of exception handling due to an address error or instruction, the I3 to I0 bits are not affected. The start address of exception service routine is then fetched from the exception handling vector table and the program starts running from that address.

7.1.3 Exception Handling Vector Table

Before exception handling starts running, the exception handling vector table must be set in memory. The exception handling vector table stores the start addresses of exception service routines. (The reset exception handling table holds the initial values of PC and SP.)

All exception sources are given different vector numbers and vector table address offsets, from which the vector table addresses are calculated. During exception handling, the start addresses of the exception service routines are fetched from the exception handling vector table, which is indicated by this vector table address.

Table 7.3 shows the vector numbers and vector table address offsets. Table 7.4 shows how vector table addresses are calculated.

Table 7.3 Exception Handling Vector Table

Exception Sources		Vector Numbers	Vector Table Address Offset
Power-on reset	PC	0	H'00000000 to H'00000003
	SP	1	H'00000004 to H'00000007
(Reserved for system use)		2	H'00000008 to H'0000000B
		3	H'0000000C to H'0000000F
General illegal instruction		4	H'00000010 to H'00000013
(Reserved for system use)		5	H'00000014 to H'00000017
Slot illegal instruction		6	H'00000018 to H'0000001B
(Reserved for system use)		7	H'0000001C to H'0000001F
		8	H'00000020 to H'00000023
CPU address error		9	H'00000024 to H'00000027
DMAC address error		10	H'00000028 to H'0000002B
Interrupts	NMI	11	H'0000002C to H'0000002F
	User break	12	H'00000030 to H'00000033
FPU exception		13	H'00000034 to H'00000037
(Reserved for system use)		14	H'00000038 to H'0000003B
Bank overflow		15	H'0000003C to H'0000003F
Bank underflow		16	H'00000040 to H'00000043

Exception Sources	Vector Numbers	Vector Table Address Offset
Integer division exception (division by zero)	17	H'00000044 to H'00000047
Integer division exception (overflow)	18	H'00000048 to H'0000004B
(Reserved for system use)	19 : 31	H'0000004C to H'0000004F : H'0000007C to H'0000007F
Trap instruction (user vector)	32 : 63	H'00000080 to H'00000083 : H'000000FC to H'000000FF
External interrupts (IRQ), on-chip peripheral module interrupts*	64 : 499	H'00000100 to H'00000103 : H'000007CC to H'000007CF

Note: * The vector numbers and vector table address offsets for each external interrupt and on-chip peripheral module interrupt are given in table 8.4 in section 8, Interrupt Controller (INTC).

Table 7.4 Calculating Exception Handling Vector Table Addresses

Exception Source	Vector Table Address Calculation
Resets	Vector table address = (vector table address offset) = (vector number) × 4
Address errors, register bank errors, interrupts, instructions	Vector table address = VBR + (vector table address offset) = VBR + (vector number) × 4

Notes: 1. For vector table address offset, see table 7.3.

2. For vector number, see table 7.3.

7.2 Resets

7.2.1 Types of Reset

A reset is the highest-priority exception handling source. This LSI supports only a power-on reset. As shown in table 7.5, the CPU state, FPU state, and on-chip peripheral module registers are initialized by a power-on reset.

Table 7.5 Exception Source Detection and Exception Handling Start Timing

	Conditions for Transition to Reset State		Internal States		
Type	**$\overline{\text{RES}}$**	**WDT Overflow**	**CPU**	**On-Chip Peripheral Modules, I/O Port**	**WTSR of WDT**
Power-on reset	Low	—	Initialized	Initialized	Initialized
	High	Power-on reset	Initialized	Initialized	Not initialized

7.2.2 Power-On Reset

(1) Power-On Reset by Means of $\overline{\text{RES}}$ Pin

When the $\overline{\text{RES}}$ pin is driven low, this LSI enters the power-on reset state. In the power-on reset state, the internal state of the CPU and all the on-chip peripheral module registers are initialized. See section 6, Reset, for details on the power-on reset and appendix A, Pin States, for the states of individual pins during the power-on reset state.

In the power-on reset state, power-on reset exception handling starts when the $\overline{\text{RES}}$ pin is first driven low for a fixed period and then returned to high. The CPU operates as follows:

1. The initial value (execution start address) of the program counter (PC) is fetched from the exception handling vector table.
2. The initial value of the stack pointer (SP) is fetched from the exception handling vector table.
3. The vector base register (VBR) is cleared to H'00000000, the interrupt mask level bits (I3 to I0) of the status register (SR) are initialized to H'F (B'1111), and the BO and CS bits are initialized. The BN bit in IBNR of the INTC is also initialized to 0.
4. The values fetched from the exception handling vector table are set in the PC and SP, and the program starts execution.

Be sure to always perform power-on reset processing when turning the system power on.

(2) Power-On Reset Initiated by WDT

When a setting is made for a power-on reset to be generated in the WDT's watchdog timer mode, and WTCNT of the WDT overflows, this LSI enters the power-on reset state.

In this case, WTSR of the WDT is not initialized by the reset signal generated by the WDT.

If a reset by the $\overline{\text{RES}}$ pin input signal occurs simultaneously with a reset by WDT overflow, the reset by the $\overline{\text{RES}}$ pin has priority, and the WOVF bit in WTSR is cleared to 0. When power-on reset exception processing is started by the WDT, the CPU operates in the same way as when a power-on reset was caused by the $\overline{\text{RES}}$ pin.

7.3 Address Errors

7.3.1 Address Error Sources

Address errors occur when instructions are fetched or data is read or written to, as shown in table 7.6.

Table 7.6 Bus Cycles and Address Errors

Bus Cycle			
Type	Bus Master	Bus Cycle Description	Address Errors
Instruction fetch*[1]	CPU	Instruction fetched from even address	None (normal)
		Instruction fetched from odd address	Address error occurs
		Instruction fetched from other than on-chip I/O register space*[2]	None (normal)
		Instruction fetched from on-chip I/O register space*[2]	Address error occurs
		Instruction fetched from external memory space in single chip mode	Address error occurs
Data read/write	CPU or DMAC	Word data accessed from even address	None (normal)
		Word data accessed from odd address	Address error occurs
		Longword data accessed from a longword boundary	None (normal)
		Longword data accessed from other than a long-word boundary	Address error occurs
		Byte or word data accessed in on-chip I/O register space*[2]	None (normal)
		Longword data accessed in 16-bit on-chip I/O register space*[2]	None (normal)
		Longword data accessed in 8-bit on-chip I/O register space*[2]	None (normal)

Bus Cycle			
Type	Bus Master	Bus Cycle Description	Address Errors
Data read/write	CPU or DMAC	External memory space accessed in single chip mode	Address error occurs

Notes: 1. If an instruction is placed within 10 bytes from the last address of the on-chip RAM space, the CPU accesses beyond the boundary to fetch the instruction, causing an address error.

2. See section 5, Address Space, for details on the on-chip I/O register space.

7.3.2 Address Error Exception Handling

When an address error occurs, the bus cycle in which the address error occurred ends*. On completion of the instruction in progress, address error exception handling starts. The CPU operates as follows:

1. The start address of exception service routine which corresponds to the address error that occurred is fetched from the exception handling vector table.
2. The status register (SR) is saved to the stack.
3. The program counter (PC) is saved to the stack. The PC value saved is the start address of the instruction to be executed after the last executed instruction.
4. After jumping to the start address of exception service routine fetched from the exception handling vector table, program execution starts. This jump is not a delayed branch.

Note: * In the case of address error related to data read/write. In the case of address error related to instruction fetch, if the bus cycle in which the address error occurred does not end by the end of the PC saving to the stack in step 3, the CPU will repeat address error exception handling until the bus cycle in which the address error occurred ends.

7.4 Register Bank Errors

7.4.1 Register Bank Error Sources

(1) Bank Overflow

In the state where saving has already been performed to all register bank areas, bank overflow occurs when acceptance of register bank overflow exception has been set by the interrupt controller (the BOVE bit in IBNR of the INTC is set to 1) and an interrupt that uses a register bank has occurred and been accepted by the CPU.

(2) Bank Underflow

Bank underflow occurs when an attempt is made to execute a RESBANK instruction while saving has not been performed to register banks.

7.4.2 Register Bank Error Exception Handling

When a register bank error occurs, register bank error exception handling starts. The CPU operates as follows:

1. The start address of exception service routine which corresponds to the register bank error that occurred is fetched from the exception handling vector table.
2. The status register (SR) is saved to the stack.
3. The program counter (PC) is saved to the stack. The PC value saved is the start address of the instruction to be executed after the last executed instruction for a bank overflow, and the start address of the executed RESBANK instruction for a bank underflow.

   To prevent multiple interrupts from occurring at a bank overflow, the interrupt priority level that caused the bank overflow is written to the interrupt mask level bits (I3 to I0) in the status register (SR).
4. After jumping to the start address of exception service routine fetched from the exception handling vector table, program execution starts. This jump is not a delayed branch.

7.5 Interrupts

7.5.1 Interrupt Sources

Table 7.7 shows the sources that start up interrupt exception handling. These are divided into NMI, user breaks, IRQ, SINT, and on-chip peripheral modules.

Table 7.7 Interrupt Sources

Type	Request Source	Number of Sources
NMI	NMI pin (external input)	1
User break	User break controller (UBC)	1
IRQ	$\overline{\text{IRQ0}}$ to $\overline{\text{IRQ7}}$ pins (external input)	8
Memory error	RAM/ROM	2
SINT	Software interrupt	15
On-chip peripheral module	Direct memory access controller (DMAC)	16
	Compare match timer (CMT)	2
	Watchdog timer (WDT)	1
	Advanced timer unit III (ATU-III)	173
	A/D converter (ADC)	26
	Serial communications interface (SCI)	20
	Renesas serial peripheral interface (RSPI)	9
	Controller area network (RCAN-TL1)	24
	Automotive direct memory access controller (A-DMAC)	2
	FlexRay	4

Each interrupt source is allocated a different vector number and vector table offset. See table 8.4 in section 8, Interrupt Controller (INTC), for more information on vector numbers and vector table address offsets.

7.5.2 Interrupt Priority Level

The interrupt priority is predetermined. When multiple interrupts occur simultaneously, the interrupt controller (INTC) determines their relative priorities and starts processing according to the results.

The priority of interrupts is expressed as priority levels 0 to 16, with priority 0 the lowest and priority 16 the highest. The NMI interrupt has priority 16 and cannot be masked, thus it is always accepted. Priority levels of the user break interrupt and the memory error interrupt are 15. The priority levels of software interrupts (SINT) are fixed to priority 15 to 1 for each source of SINT15 to SINT1. For example, the SINT15 priority level is 15, and the SINT14 priority level is 14. Priority levels of IRQ interrupts and on-chip peripheral module interrupts can be set freely using the interrupt priority registers 01 to 30 (IPR01 to IPR30) of the INTC as shown in table 7.8. The priority levels that can be set are 0 to 15. Level 16 cannot be set. See section 8.3.1, Interrupt Priority Registers 01 to 30 (IPR01 to IPR30), for details on IPR01 to IPR30.

Table 7.8 Interrupt Priority

Type	Priority Level	Comment
NMI	16	Fixed priority level. Cannot be masked.
User break	15	Fixed priority level.
IRQ	0 to 15	Set with interrupt priority register (IPR).
Memory error	15	Fixed priority level.
SINT15 to SINT1	15 to 1	Fixed priority level.
On-chip peripheral module	0 to 15	Set with interrupt priority register (IPR).

7.5.3 Interrupt Exception Handling

When an interrupt occurs, its priority level is ascertained by the interrupt controller (INTC). NMI is always accepted, but other interrupts are only accepted if they have a priority level higher than the priority level set in the interrupt mask level bits (I3 to I0) of the status register (SR).

When an interrupt is accepted, interrupt exception handling starts. In interrupt exception handling, the CPU fetches the exception service routine start address which corresponds to the accepted interrupt from the exception handling vector table, and saves SR and the program counter (PC) to the stack. In the case of interrupt exception handling other than NMI or user break with usage of the register banks enabled, general registers R0 to R14, control register GBR, system registers MACH, MACL, and PR, and the vector table address offset of the interrupt exception handling to be executed are saved in the register banks. In the case of exception handling due to an address error, NMI interrupt, user break interrupt, or instruction, saving is not performed to the register banks. If saving has been performed to all register banks (0 to 14), automatic saving to the stack is performed instead of register bank saving. In this case, an interrupt controller setting must have been made so that a register bank overflow exception is not accepted (the BOVE bit in IBNR of the INTC is 0). If a setting to accept a register bank overflow exception has been made (the BOVE bit in IBNR of the INTC is 1), the register bank overflow exception occurs. Next, the priority level value of the accepted interrupt is written to the I3 to I0 bits in SR. For NMI, however, the priority level is 16, but the value set in the I3 to I0 bits is H'F (level 15). Then, after jumping to the start address fetched from the exception handling vector table, program execution starts. The jump that occurs is not a delayed branch. See section 8.6, Operation, for further details on interrupt exception handling.

7.6 Exceptions Triggered by Instructions

7.6.1 Types of Exceptions Triggered by Instructions

Exception handling can be triggered by trap instructions, slot illegal instructions, general illegal instructions, integer division exceptions, and FPU exceptions as shown in table 7.9.

Table 7.9 Types of Exceptions Triggered by Instructions

Type	Source Instruction	Comment
Trap instruction	TRAPA	
Slot illegal instruction	Undefined code placed immediately after a delayed branch instruction (delay slot) (including FPU instructions and FPU-related CPU instructions in FPU module standby state), instructions that rewrite the PC, 32-bit instructions, RESBANK instruction, DIVS instruction, and DIVU instruction	Delayed branch instructions: JMP, JSR, BRA, BSR, RTS, RTE, BF/S, BT/S, BSRF, and BRAF Instructions that rewrite the PC: JMP, JSR, BRA, BSR, RTS, RTE, BT, BF, TRAPA, BF/S, BT/S, BSRF, BRAF, JSR/N, and RTV/N 32-bit instructions: BAND.B, BANDNOT.B, BCLR.B, BLD.B, BLDNOT.B, BOR.B, BORNOT.B, BSET.B, BST.B, BXOR.B, FMOV.S@disp12, FMOV.D@disp12, MOV.B@disp12, MOV.W@disp12, MOV.L@disp12, MOVI20, MOVI20S, MOVU.B, MOVU.W.
General illegal instruction	Undefined code anywhere besides in a delay slot (including FPU instructions and FPU-related CPU instructions in FPU module standby status)	
Integer division exception	Division by zero	DIVU, DIVS
	Negative maximum value ÷ (−1)	DIVS
FPU exceptions	Starts when detecting invalid operation exception defined by IEEE754, division-by-zero exception, overflow, underflow, or inexact exception.	FADD, FSUB, FMUL, FDIV, FMAC, FCMP/EQ, FCMP/GT, FNEG, FABS, FLOAT, FTRC, FCNVDS, FCNVSD, FSQRT

7.6.2 Trap Instructions

When a TRAPA instruction is executed, trap instruction exception handling starts. The CPU operates as follows:

1. The start address of exception service routine which corresponds to the vector number specified in the TRAPA instruction is fetched from the exception handling vector table.
2. The status register (SR) is saved to the stack.
3. The program counter (PC) is saved to the stack. The PC value saved is the start address of the instruction to be executed after the TRAPA instruction.
4. After jumping to the start address of exception service routine fetched from the exception handling vector table, program execution starts. This jump is not a delayed branch.

7.6.3 Slot Illegal Instructions

An instruction placed immediately after a delayed branch instruction is said to be placed in a delay slot. When the instruction placed in the delay slot is undefined code (including FPU instructions and FPU-related CPU instructions in FPU module standby state), an instruction that rewrites the PC, a 32-bit instruction, an RESBANK instruction, a DIVS instruction, or a DIVU instruction, slot illegal exception handling starts when such kind of instruction is decoded. The CPU operates as follows:

1. The start address of exception service routine is fetched from the exception handling vector table.
2. The status register (SR) is saved to the stack.
3. The program counter (PC) is saved to the stack. The PC value saved is the jump address of the delayed branch instruction immediately before the undefined code (including FPU instructions and FPU-related CPU instructions in FPU module standby state), the instruction that rewrites the PC, the 32-bit instruction, the RESBANK instruction, the DIVS instruction, or the DIVU instruction.
4. After jumping to the start address of exception service routine fetched from the exception handling vector table, program execution starts. This jump is not a delayed branch.

7.6.4 General Illegal Instructions

When undefined code placed anywhere other than immediately after a delayed branch instruction (i.e., in a delay slot) (including FPU instructions and FPU-related CPU instructions in FPU module standby state) is decoded, general illegal instruction exception handling starts. The CPU handles general illegal instructions in the same way as slot illegal instructions. Unlike processing of slot illegal instructions, however, the program counter value stored is the start address of the undefined code.

7.6.5 Integer Division Exceptions

When an integer division instruction performs division by zero or the result of integer division overflows, integer division instruction exception handling starts. The instructions that may become the source of division-by-zero exception are DIVU and DIVS. The only source instruction of overflow exception is DIVS, and overflow exception occurs only when the negative maximum value is divided by −1. The CPU operates as follows:

1. The start address of exception service routine which corresponds to the integer division instruction exception that occurred is fetched from the exception handling vector table.
2. The status register (SR) is saved to the stack.
3. The program counter (PC) is saved to the stack. The PC value saved is the start address of the integer division instruction at which the exception occurred.
4. After jumping to the start address of exception service routine fetched from the exception handling vector table, program execution starts. This jump is not a delayed branch.

7.6.6 FPU Exceptions

An FPU exception handling is generated when the V, Z, O, U or I bit in the FPU enable field (Enable) of the floating point status/control register (FPSCR) is set. This indicates the occurrence of an invalid operation exception defined by the IEEE standard 754, a division-by-zero exception, overflow (in the case of an instruction for which this is possible), underflow (in the case of an instruction for which this is possible), or inexact exception (in the case of an instruction for which this is possible).

The instructions that may cause the generation of an FPU exception handling are FADD, FSUB, FMUL, FDIV, FMAC, FCMP/EQ, FCMP/GT, FLOAT, FTRC, FCNVDS, FCNVSD, and FSQRT.

An FPU exception is generated only when the corresponding FPU exception enable bit (Enable) is set. When the FPU detects an exception source by the floating point operation instructions, FPU operation is halted and a generation of FPU exception handling is reported to the CPU. When exception handling is started, the CPU operations are as follows.

1. The start address of the exception service routine which corresponds to the FPU exception handling that occurred is fetched from the exception handling vector table.
2. The status register (SR) is saved to the stack.
3. The program counter (PC) is saved to the stack. The PC value saved is the start address of the instruction to be executed after the last executed instruction.
4. After jumping to the address fetched from the exception handling vector table, program execution starts. This jump is not a delayed branch.

The FPU exception flag field (Flag) of FPSCR is always updated regardless of whether or not an FPU exception handling has been accepted, and remains set until explicitly cleared by the user through an instruction. The FPU exception source field (Cause) of FPSCR changes each time an floating point operation instruction is executed.

When the V bit in the FPU exception enable field (Enable) of FPSCR is set and the QIS bit in FPSCR is also set, FPU exception is generated when qNAN or ±∞ is input to a floating point operation instruction source.

7.7 When Exception Sources Are Not Accepted

When an address error, FPU exception, register bank error (overflow), or interrupt is generated immediately after a delayed branch instruction, it is sometimes not accepted immediately but stored instead, as shown in table 7.10. When this happens, it will be accepted when an instruction that can accept the exception is decoded.

Table 7.10 Exception Source Generation Immediately after Delayed Branch Instruction

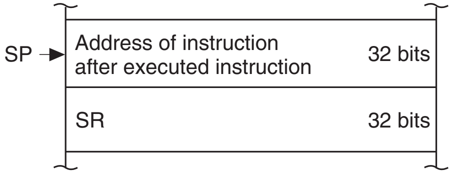
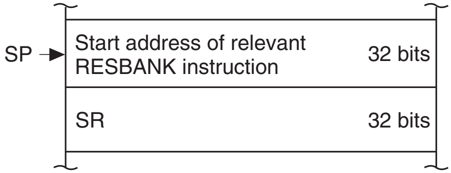
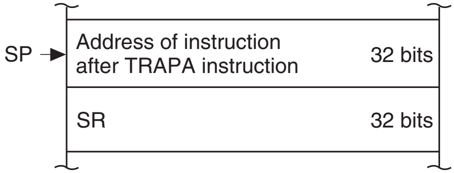
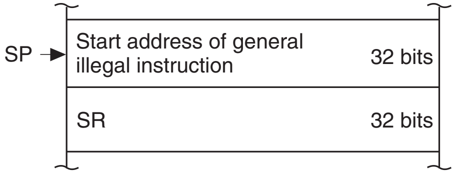
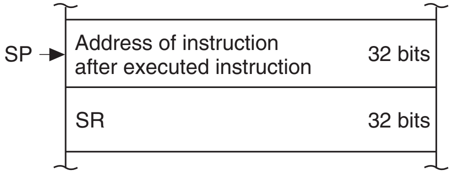
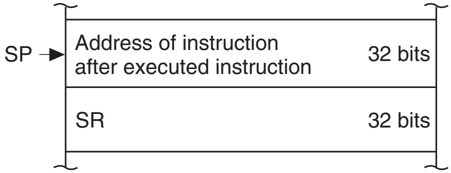
	Exception Source			
Point of Occurrence	**Address Error**	**FPU Exception**	**Register Bank Error (Overflow)**	**Interrupt**
Immediately after a delayed branch instruction*	Not accepted	Not accepted	Not accepted	Not accepted

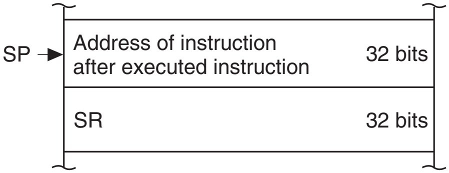
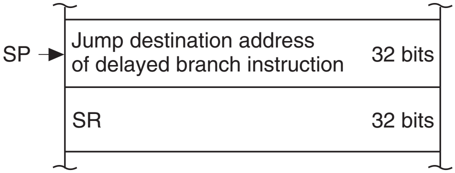
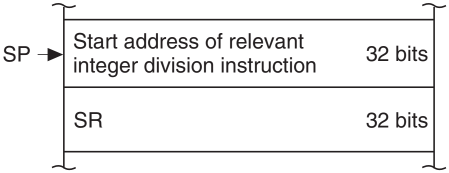
Note: * Delayed branch instructions: JMP, JSR, BRA, BSR, RTS, RTE, BF/S, BT/S, BSRF, and BRAF.

7.8 Stack Status after Exception Handling Ends

The state of the stack after exception handling ends is as shown in table 7.11.

Table 7.11 Stack Status After Exception Handling Ends

Exception Type	Stack Status
Address error	 SP → Address of instruction after executed instruction 32 bits SR 32 bits
Register bank error (underflow)	 SP → Start address of relevant RESBANK instruction 32 bits SR 32 bits
Trap instruction	 SP → Address of instruction after TRAPA instruction 32 bits SR 32 bits
General illegal instruction	 SP → Start address of general illegal instruction 32 bits SR 32 bits
Interrupt	 SP → Address of instruction after executed instruction 32 bits SR 32 bits
Register bank error (overflow)	 SP → Address of instruction after executed instruction 32 bits SR 32 bits

Exception Type	Stack Status
FPU exception	 SP Address of instruction after executed instruction 32 bits SR 32 bits
Slot illegal instruction	 SP Jump destination address of delayed branch instruction 32 bits SR 32 bits
Integer division instruction	 SP Start address of relevant integer division instruction 32 bits SR 32 bits

7.9 Usage Notes

7.9.1 Value of Stack Pointer (SP)

The value of the stack pointer must always be a multiple of four. If it is not, an address error will occur when the stack is accessed during exception handling.

7.9.2 Value of Vector Base Register (VBR)

The value of the vector base register must always be a multiple of four. If it is not, an address error will occur when the stack is accessed during exception handling.

7.9.3 Address Errors Caused by Stacking of Address Error Exception Handling

When the stack pointer is not a multiple of four, an address error will occur during stacking of the exception handling (interrupts, etc.) and address error exception handling will start up as soon as the first exception handling is ended. Address errors will then also occur in the stacking for this address error exception handling. To ensure that address error exception handling does not go into an endless loop, no address errors are accepted at that point. This allows program control to be shifted to the address error exception service routine and enables error processing.

When an address error occurs during exception handling stacking, the stacking bus cycle (write) is executed. During stacking of the status register (SR) and program counter (PC), the SP is decremented by 4 for both, so the value of SP will not be a multiple of four after the stacking either. The address value output during stacking is the SP value, so the address where the error occurred is itself output. This means the write data stacked will be undefined.

Section 8 Interrupt Controller (INTC)

The interrupt controller (INTC) ascertains the priority of interrupt sources and controls interrupt requests to the CPU. The priority of each interrupt can be set by the INTC registers, and interrupts are processed according to the user-set priority.

8.1 Features

- 16 levels of interrupt priority can be set

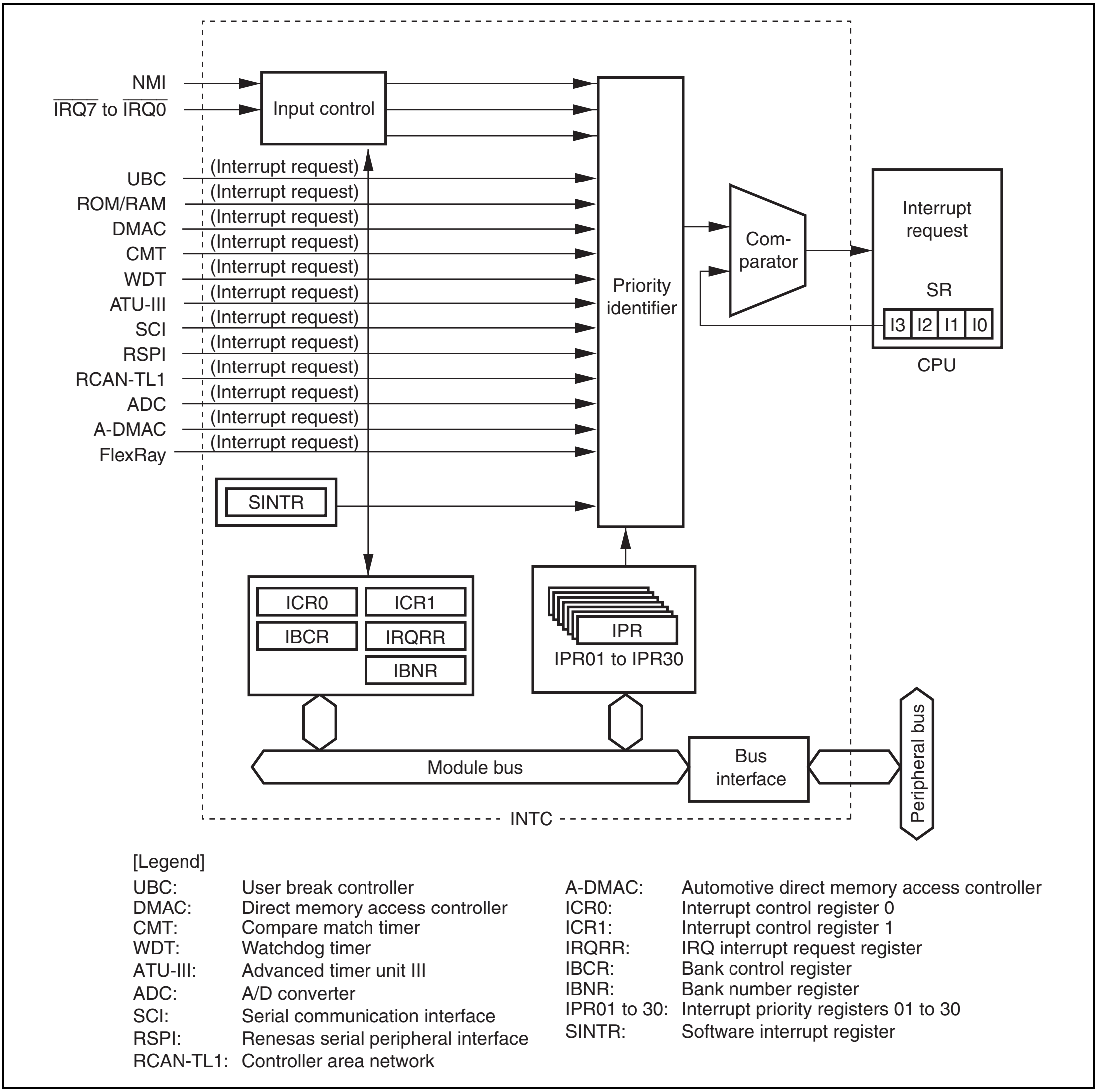
  By setting the thirty interrupt priority registers, the priorities of IRQ interrupts and on-chip peripheral module interrupts can be selected from 16 levels for request sources.
- NMI noise canceller function

  An NMI input-level bit indicates the NMI pin state. By reading this bit in the interrupt exception handler, the pin state can be checked, enabling it to be used as the noise canceller function.
- Register banks

  This LSI has register banks that enable register saving and restoring required in the interrupt processing to be performed at high speed.
- Software interrupt (SINT)

  By setting the software interrupt register, an interrupt with a given priority can be generated from a program.

Figure 8.1 shows a block diagram of the INTC.



[Legend]

UBC:	User break controller	A-DMAC:	Automotive direct memory access controller
DMAC:	Direct memory access controller	ICR0:	Interrupt control register 0
CMT:	Compare match timer	ICR1:	Interrupt control register 1
WDT:	Watchdog timer	IRQRR:	IRQ interrupt request register
ATU-III:	Advanced timer unit III	IBCR:	Bank control register
ADC:	A/D converter	IBNR:	Bank number register
SCI:	Serial communication interface	IPR01 to 30:	Interrupt priority registers 01 to 30
RSPI:	Renesas serial peripheral interface	SINTR:	Software interrupt register
RCAN-TL1:	Controller area network		

Figure 8.1 Block Diagram of INTC

8.2 Input/Output Pins

Table 8.1 shows the pin configuration of the INTC.

Table 8.1 Pin Configuration

Pin Name	Symbol	I/O	Function
Nonmaskable interrupt input pin	NMI	Input	Input pin for the nonmaskable interrupt request signal
Interrupt request input pins	$\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$	Input	Input pins for maskable interrupt request signals

8.3 Register Descriptions

The INTC has the following registers. These registers are used to set the interrupt priorities and control detection of the external interrupt input signal.

Table 8.2 Register Configuration

Register Name	Symbol	R/W	Initial Value	Address	Access Size
Interrupt control register 0	ICR0	R/W	*1	H'FFFE0800	16, 32
Interrupt control register 1	ICR1	R/W	H'0000	H'FFFE0802	16, 32
IRQ interrupt request register	IRQRR	R/(W)*2	H'0000	H'FFFE0806	8, 16, 32
Bank control register	IBCR	R/W	H'0000	H'FFFE080C	16, 32
Bank number register	IBNR	R/W	H'0000	H'FFFE080E	16, 32
Software interrupt register 1	SINTR1	R/W	H'00	H'FFFE0810	8, 16, 32
Software interrupt register 2	SINTR2	R/W	H'00	H'FFFE0811	8, 16, 32*3
Software interrupt register 3	SINTR3	R/W	H'00	H'FFFE0812	8, 16, 32
Software interrupt register 4	SINTR4	R/W	H'00	H'FFFE0813	8, 16, 32*3
Software interrupt register 5	SINTR5	R/W	H'00	H'FFFE0814	8, 16, 32
Software interrupt register 6	SINTR6	R/W	H'00	H'FFFE0815	8, 16, 32*3
Software interrupt register 7	SINTR7	R/W	H'00	H'FFFE0816	8, 16, 32
Software interrupt register 8	SINTR8	R/W	H'00	H'FFFE0817	8, 16, 32*3
Interrupt priority register 01	IPR01	R/W	H'0000	H'FFFE0818	16, 32

Register Name	Symbol	R/W	Initial Value	Address	Access Size
Interrupt priority register 02	IPR02	R/W	H'0000	H'FFFE081A	16, 32
Software interrupt register 9	SINTR9	R/W	H'00	H'FFFE0828	8, 16, 32
Software interrupt register 10	SINTR10	R/W	H'00	H'FFFE0829	8, 16, 32*[3]
Software interrupt register 11	SINTR11	R/W	H'00	H'FFFE082A	8, 16, 32
Software interrupt register 12	SINTR12	R/W	H'00	H'FFFE082B	8, 16, 32*[3]
Software interrupt register 13	SINTR13	R/W	H'00	H'FFFE082C	8, 16, 32
Software interrupt register 14	SINTR14	R/W	H'00	H'FFFE082D	8, 16, 32*[3]
Software interrupt register 15	SINTR15	R/W	H'00	H'FFFE082E	8, 16, 32
Interrupt priority register 03	IPR03	R/W	H'0000	H'FFFE0C00	16, 32
Interrupt priority register 04	IPR04	R/W	H'0000	H'FFFE0C02	16, 32
Interrupt priority register 05	IPR05	R/W	H'0000	H'FFFE0C04	16, 32
Interrupt priority register 06	IPR06	R/W	H'0000	H'FFFE0C06	16, 32
Interrupt priority register 07	IPR07	R/W	H'0000	H'FFFE0C08	16, 32
Interrupt priority register 08	IPR08	R/W	H'0000	H'FFFE0C0A	16, 32
Interrupt priority register 09	IPR09	R/W	H'0000	H'FFFE0C0C	16, 32
Interrupt priority register 10	IPR10	R/W	H'0000	H'FFFE0C0E	16, 32
Interrupt priority register 11	IPR11	R/W	H'0000	H'FFFE0C10	16, 32
Interrupt priority register 12	IPR12	R/W	H'0000	H'FFFE0C12	16, 32
Interrupt priority register 13	IPR13	R/W	H'0000	H'FFFE0C14	16, 32
Interrupt priority register 14	IPR14	R/W	H'0000	H'FFFE0C16	16, 32
Interrupt priority register 15	IPR15	R/W	H'0000	H'FFFE0C18	16, 32
Interrupt priority register 16	IPR16	R/W	H'0000	H'FFFE0C1A	16, 32
Interrupt priority register 17	IPR17	R/W	H'0000	H'FFFE0C1C	16, 32
Interrupt priority register 18	IPR18	R/W	H'0000	H'FFFE0C1E	16, 32
Interrupt priority register 19	IPR19	R/W	H'0000	H'FFFE0C20	16, 32
Interrupt priority register 20	IPR20	R/W	H'0000	H'FFFE0C22	16, 32
Interrupt priority register 21	IPR21	R/W	H'0000	H'FFFE0C24	16, 32
Interrupt priority register 22	IPR22	R/W	H'0000	H'FFFE0C26	16, 32
Interrupt priority register 23	IPR23	R/W	H'0000	H'FFFE0C28	16, 32
Interrupt priority register 24	IPR24	R/W	H'0000	H'FFFE0C2A	16, 32

Register Name	Symbol	R/W	Initial Value	Address	Access Size
Interrupt priority register 25	IPR25	R/W	H'0000	H'FFFE0C2C	16, 32
Interrupt priority register 26	IPR26	R/W	H'0000	H'FFFE0C2E	16, 32
Interrupt priority register 27	IPR27	R/W	H'0000	H'FFFE0C30	16, 32
Interrupt priority register 28	IPR28	R/W	H'0000	H'FFFE0C32	16, 32
Interrupt priority register 29	IPR29	R/W	H'0000	H'FFFE0C34	16, 32
Interrupt priority register 30	IPR30	R/W	H'0000	H'FFFE0C70	16, 32

Notes: 1. H'8000 when the NMI pin is high and H'0000 when low.

2. Only 0 can be written after reading 1, to clear the flag.

3. In cases of access to an odd address, select eight bits as the unit of access.

8.3.1 Interrupt Priority Registers 01 to 30 (IPR01 to IPR30)

IPR01 to IPR30 are 16-bit readable/writable registers in which priority levels from 0 to 15 are set for IRQ interrupts and on-chip peripheral module interrupts. Table 8.3 shows the correspondence between the interrupt request sources and the bits in IPR01 to IPR30.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8.3 Interrupt Request Sources and IPR01 to IPR30

Register Name	Bits 15 to 12	Bits 11 to 8	Bits 7 to 4	Bits 3 to 0
Interrupt priority register 01	IRQ0	IRQ1	IRQ2	IRQ3
Interrupt priority register 02	IRQ4	IRQ5	IRQ6	IRQ7
Interrupt priority register 03	DMAC0	DMAC1	DMAC2	DMAC3
Interrupt priority register 04	DMAC4	DMAC5	DMAC6	DMAC7
Interrupt priority register 05	CMT0	CMT1	Reserved	WDT
Interrupt priority register 06	ATU-A (ICIA0, ICIA1)	ATU-A (ICIA2, ICIA3)	ATU-A (ICIA4, ICIA5)	ATU-A (OVIA)

Register Name	Bits 15 to 12	Bits 11 to 8	Bits 7 to 4	Bits 3 to 0
Interrupt priority register 07	ATU-B (CMIB0, CMIB1)	ATU-B (CMIB6, ICIB0)	ATU-C0 (IMIC00 to IMIC03)	ATU-C0 (OVIC0)
Interrupt priority register 08	ATU-C1 (IMIC10 to IMIC13)	ATU-C1 (OVIC1)	ATU-C2 (IMIC20 to IMIC23)	ATU-C2 (OVIC2)
Interrupt priority register 09	ATU-C3 (IMIC30 to IMIC33)	ATU-C3 (OVIC3)	ATU-C4 (IMIC40 to IMIC43)	ATU-C4 (OVIC4)
Interrupt priority register 10	ATU-D0 (CMID00 to CMID03)	ATU-D0 (OVI1D0, OVI2D0)	ATU-D0 (UDID00 to UDID03)	ATU-D1 (CMID10 to CMID13)
Interrupt priority register 11	ATU-D1 (OVI1D1, OVI2D1)	ATU-D1 (UDID10 to UDID13)	ATU-D2 (CMID20 to CMID23)	ATU-D2 (OVI1D2, OVI2D2)
Interrupt priority register 12	ATU-D2 (UDID20 to UDID23)	ATU-D3 (CMID30 to CMID33)	ATU-D3 (OVI1D3, OVI2D3)	ATU-D3 (UDID30 to UDID33)
Interrupt priority register 13	Reserved	Reserved	Reserved	Reserved
Interrupt priority register 14	Reserved	Reserved	ATU-E0 (CMIE00 to CMIE03)	ATU-E1 (CMIE10 to CMIE13)
Interrupt priority register 15	ATU-E2 (CMIE20 to CMIE23)	ATU-E3 (CMIE30 to CMIE33)	ATU-E4 (CMIE40 to CMIE43)	ATU-E5 (CMIE50 to CMIE53)
Interrupt priority register 16	ATU-F (ICIF0 to ICIF3)	ATU-F (ICIF4 to ICIF7)	ATU-F (ICIF8 to ICIF11)	ATU-F (ICIF12 to ICIF15)
Interrupt priority register 17	ATU-F (ICIF16 to ICIF19)	ATU-F (ICIF20 to ICIF23)	ATU-F (ICIF24 to ICIF27)	Reserved
Interrupt priority register 18	ATU-F (OVIF0 to OVIF3)	ATU-F (OVIF4 to OVIF7)	ATU-F (OVIF8 to OVIF11)	ATU-F (OVIF12 to OVIF15)
Interrupt priority register 19	ATU-F (OVIF16 to OVIF19)	ATU-F (OVIF20 to OVIF23)	ATU-F (OVIF24 to OVIF27)	Reserved
Interrupt priority register 20	ATU-G (CMIG0 to CMIG3)	ATU-G (CMIG4, CMIG5)	ATU-H (CMIH)	Reserved
Interrupt priority register 21	ATU-J (DFIJ0, DFIJ1)	ATU-J (OVIJ0, OVIJ1)	ATU-J (DOVIJ0, DOVIJ1)	ATU-E6 (CMIE60 to CMIE63)

Register Name	Bits 15 to 12	Bits 11 to 8	Bits 7 to 4	Bits 3 to 0
Interrupt priority register 22	ADC (ADI0)	ADC (ADI1)	ADC (ADID0 to ADID3)	ADC (ADID4 to ADID7)
Interrupt priority register 23	ADC (ADID8 to ADID11)	ADC (ADID12 to ADID15)	ADC (ADID40)	ADC (ADID41)
Interrupt priority register 24	ADC (ADID42)	ADC (ADID43)	ADC (ADID44)	ADC (ADID45)
Interrupt priority register 25	ADC (ADID46)	ADC (ADID47)	Reserved	Reserved
Interrupt priority register 26	SCI_A	SCI_B	SCI_C	SCI_D
Interrupt priority register 27	SCI_E	RSPI_A	RSPI_B	RSPI_C
Interrupt priority register 28	RCAN_A	RCAN_B	RCAN_C	RCAN_D
Interrupt priority register 29	A-DMAC (TE74)	A-DMAC (TE75)	Reserved	Reserved
Interrupt priority register 30	FlexRay FR0	FlexRay FR1	FlexRay FRT0	FlexRay FRT1

As shown in table 8.3, by setting the 4-bit groups (bits 15 to 12, bits 11 to 8, bits 7 to 4, and bits 3 to 0) with values from H'0 (0000) to H'F (1111), the priority of each corresponding interrupt is set. Setting of H'0 means priority level 0 (the lowest level) and H'F means priority level 15 (the highest level).

IPR01 to IPR30 are initialized to H'0000 by a power-on reset.

8.3.2 Interrupt Control Register 0 (ICR0)

ICR0 is a 16-bit register that sets the input signal detection mode for the external interrupt input pin NMI, and indicates the input level at the NMI pin. ICR0 is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NMIL	-	-	-	-	-	-	NMIE	-	-	-	-	-	-	-	-
Initial value:	*	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R

Note: * 1 when the NMI pin is high, and 0 when the NMI pin is low.

Bit	Bit Name	Initial Value	R/W	Description
15	NMIL	*	R	NMI Input Level Sets the level of the signal input at the NMI pin. The NMI pin level can be obtained by reading this bit. This bit cannot be modified. 0: Low level is input to NMI pin 1: High level is input to NMI pin
14 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	NMIE	0	R/W	NMI Edge Select Selects whether the falling or rising edge of the interrupt request signal on the NMI pin is detected. 0: Interrupt request is detected on falling edge of NMI input 1: Interrupt request is detected on rising edge of NMI input
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: * 1 when the NMI pin is high, and 0 when the NMI pin is low.

8.3.3 Interrupt Control Register 1 (ICR1)

ICR1 is a 16-bit register that specifies the detection mode for external interrupt input pins $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$ individually: low level, falling edge, rising edge, or both edges. ICR1 is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IRQ71S	IRQ70S	IRQ61S	IRQ60S	IRQ51S	IRQ50S	IRQ41S	IRQ40S	IRQ31S	IRQ30S	IRQ21S	IRQ20S	IRQ11S	IRQ10S	IRQ01S	IRQ00S
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	IRQ71S	0	R/W	IRQ Sense Select
14	IRQ70S	0	R/W	These bits select whether interrupt signals corresponding to pins $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$ are detected by a low level, falling edge, rising edge, or both edges. 00: Interrupt request is detected on low level of $\overline{\text{IRQn}}$ input 01: Interrupt request is detected on falling edge of $\overline{\text{IRQn}}$ input 10: Interrupt request is detected on rising edge of $\overline{\text{IRQn}}$ input 11: Interrupt request is detected on both edges of $\overline{\text{IRQn}}$ input
13	IRQ61S	0	R/W	
12	IRQ60S	0	R/W	
11	IRQ51S	0	R/W	
10	IRQ50S	0	R/W	
9	IRQ41S	0	R/W	
8	IRQ40S	0	R/W	
7	IRQ31S	0	R/W	
6	IRQ30S	0	R/W	
5	IRQ21S	0	R/W	
4	IRQ20S	0	R/W	
3	IRQ11S	0	R/W	
2	IRQ10S	0	R/W	
1	IRQ01S	0	R/W	
0	IRQ00S	0	R/W	

[Legend]

n = 7 to 0

8.3.4 IRQ Interrupt Request Register (IRQRR)

IRQRR is a 16-bit register that indicates interrupt requests from external input pins $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$. If edge detection is set for the IRQ7 to IRQ0 interrupts, writing 0 to the IRQ7F to IRQ0F bits after reading IRQ7F to IRQ0F = 1 cancels the retained interrupts.

IRQRR is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only 0 can be written to clear the flag after 1 is read.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
7	IRQ7F	0	R/(W)*	IRQ Interrupt Request These bits indicate the status of the IRQ7 to IRQ0 interrupt requests. Level detection: 0: IRQn interrupt request has not occurred [Clearing condition] • $\overline{\text{IRQn}}$ input is high 1: IRQn interrupt has occurred [Setting condition] • $\overline{\text{IRQn}}$ input is low Edge detection: 0: IRQn interrupt request is not detected [Clearing conditions] • Cleared by reading IRQnF while IRQnF = 1, then writing 0 to IRQnF • Cleared by executing IRQn interrupt exception handling 1: IRQn interrupt request is detected [Setting condition] • Edge corresponding to IRQn1S or IRQn0S of ICR1 has occurred at $\overline{\text{IRQn}}$ pin
6	IRQ6F	0	R/(W)*	
5	IRQ5F	0	R/(W)*	
4	IRQ4F	0	R/(W)*	
3	IRQ3F	0	R/(W)*	
2	IRQ2F	0	R/(W)*	
1	IRQ1F	0	R/(W)*	
0	IRQ0F	0	R/(W)*	

[Legend]

n = 7 to 0

8.3.5 Bank Control Register (IBCR)

IBCR is a 16-bit register that enables or disables use of register banks for each interrupt priority level. IBCR is initialized to H'0000 by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
15	E15	0	R/W	Enable These bits enable or disable use of register banks for interrupt priority levels 15 to 1. However, use of register banks is always disabled for the user break interrupts. 0: Use of register banks is disabled 1: Use of register banks is enabled
14	E14	0	R/W	
13	E13	0	R/W	
12	E12	0	R/W	
11	E11	0	R/W	
10	E10	0	R/W	
9	E9	0	R/W	
8	E8	0	R/W	
7	E7	0	R/W	
6	E6	0	R/W	
5	E5	0	R/W	
4	E4	0	R/W	
3	E3	0	R/W	
2	E2	0	R/W	
1	E1	0	R/W	
0	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

8.3.6 Bank Number Register (IBNR)

IBNR is a 16-bit register that enables or disables use of register banks and register bank overflow exception. IBNR also indicates the bank number to which saving is performed next through the bits BN3 to BN0.

IBNR is initialized to H'0000 by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BE[1:0]		BOVE	-	-	-	-	-	-	-	-	-	BN[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15, 14	BE[1:0]	00	R/W	Register Bank Enable These bits enable or disable use of register banks. 00: Use of register banks is disabled for all interrupts. The setting of IBCR is ignored. 01: Use of register banks is enabled for all interrupts except NMI and user break. The setting of IBCR is ignored. 10: Reserved (setting prohibited) 11: Use of register banks is controlled by the setting of IBCR.
13	BOVE	0	R/W	Register Bank Overflow Enable Enables of disables register bank overflow exception. 0: Generation of register bank overflow exception is disabled 1: Generation of register bank overflow exception is enabled
12 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
3 to 0	BN[3:0]	0000	R	Bank Number These bits indicate the bank number to which saving is performed next. When an interrupt using register banks is received, saving is performed to the register bank indicated by these bits, and BN is incremented by 1. After BN is decremented by 1 due to execution of a RESBANK (restore from register bank) instruction, restoring from the register bank is performed.

8.3.7 Software Interrupt Registers 1 to 15 (SINTR1 to SINTR15)

SINTR1 to SINTR15 are 8-bit registers that control software interrupts 1 to 15 (SINT1 to SINT15). Writing H'01 to this register generates the software interrupts 1 to 15 (SINT1 to SINT15). Writing H'00 while handling the generated interrupts clears the interrupt sources. When SINTR1 to SINTR15 are read, the current register value is read.

SINTR1 to SINTR15 are initialized to H'00 by a power-on reset.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	SINTC
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	SINTC	0	R/W	Software Interrupt Request These bits generate software interrupts 1 to 15 (SINT1 to SINT15). [Read operation] Current bit value is read. [Write operation] Writing 1: Generates interrupts Writing 1 to this bit is prohibited when the SINTC bit is 1 Writing 0: Clears interrupt sources

8.4 Interrupt Sources

There are six types of interrupt sources: the NMI interrupt, user break, IRQ, memory error interrupt, software interrupts (SINT), and interrupts from on-chip peripheral modules. Each interrupt has a priority level (0 to 16). A level of 0 corresponds to the lowest and a level of 16 corresponds to the highest. When the level is set to 0, the interrupt is masked at all times.

8.4.1 NMI Interrupt

The NMI interrupt has a priority level of 16 and is received at all times. The edge of the NMI signal is detected as an NMI interrupt and the NMI edge select bit (NMIE) in interrupt control register 0 (ICR0) selects whether the rising edge or falling edge is detected.

Although the priority level of the NMI interrupt is 16, the interrupt mask level bits (I3 to I0) in the status register (SR) are set to level 15 in the NMI interrupt exception handler.

8.4.2 User Break Interrupt

A user break interrupt occurs when a break condition set in the user break controller (UBC) is satisfied and has a priority level of 15. Bits I3 to I0 in SR is set to level 15 in the user break interrupt exception handler. For details on the user break interrupt, see section 9, User Break Controller (UBC).

8.4.3 IRQ Interrupts

An IRQ interrupt is input on pins $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$. The low level, falling edge, rising edge, or both edge of the IRQ signals is detected and the edge to be detected can be selected individually for each pin by the IRQ sense select bits (IRQ71S to IRQ01S and IRQ70S to IRQ00S) in interrupt control register 1 (ICR1). The priority level can be set individually in a range from 0 to 15 for each pin by interrupt priority registers 01 and 02 (IPR01 and IPR02).

When using the low-level sensing for IRQ interrupts, an interrupt is requested to the INTC while signals $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$ are driven low. When the signals are driven high, the interrupt stops to be requested. Whether or not an IRQ interrupt has occurred can be checked by reading the IRQ interrupt request bits (IRQ7R to IRQ0R) in the IRQ interrupt request register (IRQRR).

When using a edge sensing for IRQ interrupts, an interrupt is requested to the INTC when an interrupt request is detected due to changes in signals $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$. The interrupt request is held until the interrupt is received and whether or not an interrupt has requested can be checked by reading bits IRQ7R to IRQ0R in IRQRR. The request can be cleared by writing 0 to these bits after reading them as 1.

In the IRQ interrupt exception handler, bits I3 to I0 in SR are set to the priority level of the received IRQ interrupt.

Before execution of the RTE instruction to exit the IRQ interrupt exception handler, confirm that the interrupt has been cleared by reading the IRQ interrupt request register (IRQRR). Otherwise, the interrupt is requested again erroneously.

8.4.4 Memory Error Interrupt

For details on the sources generating a memory error, see section 26, ROM, and section 30, RAM.

8.4.5 Software Interrupts (SINT)

A software interrupt (SINT) is generated by setting SINTR1 to SINTR15 by software. The interrupt priority levels of the software interrupts (SINT) are fixed as shown in table 8.4. In the SINT interrupt exception handles, bits I3 to I0 in SR is set to the priority level of the received software interrupt (SINT).

8.4.6 On-Chip Peripheral Module Interrupts

The following on-chip peripheral modules can generate on-chip peripheral module interrupts.

- Direct memory access controller (DMAC)
- Compare match timer (CMT)
- Watchdog timer (WDT)
- Advanced timer unit III (ATU-III)
- A/D converter (ADC)
- Serial communications interface (SCI)
- Renesas serial peripheral interface (RSPI)
- FlexRay
- Controller area network (RCAN-TL1)
- Automotive direct memory access controller (A-DMAC)

Since each source is assigned to a unique interrupt vector, the source does not need to be identified in the interrupt exception handler. A priority level in a range from 0 to 15 can be set for each module by interrupt priority registers 03 to 30 (IPR03 to IPR30). In the exception handler for the on-chip peripheral module interrupt, bits I3 to I0 in SR is set to the priority level of the received on-chip peripheral module interrupt.

8.5 Interrupt Exception Handling Vector Table and Priority

Table 8.4 lists interrupt sources and their vector numbers, vector table address offsets, and interrupt priority levels.

Each interrupt source is assigned to a unique vector number and a unique vector table address offset. Vector table addresses are calculated from the vector numbers and vector table address offsets. In the interrupt exception handler, the start address is fetched from the vector table pointed to by the vector table address. For details of calculation of the vector table address, see table 7.4, Calculating Exception Handling Vector Table Addresses, in section 7, Exception Handling.

The priorities of IRQ interrupts and on-chip peripheral module interrupts can be set freely between 0 and 15 for each pin or module by setting interrupt priority registers 01 to 30 (IPR01 to IPR30). However, if two or more interrupts specified by the same IPR among IPR03 to IPR30 occur, the interrupts are processed according to the priority levels defined as shown in the Priority within IPR Setting Unit columns in table 8.4, and the priority levels cannot be changed. The priority levels of IRQ and on-chip peripheral module interrupts are initialized to 0 by a power-on reset. If the same priority level is assigned to two or more interrupt sources and interrupts from those sources occur simultaneously, they are processed according to the default priority levels shown in the Default Priority columns in table 8.4.

Table 8.4 Interrupt Exception Handling Vectors and Priority Levels

		Interrupt Vector					
Interrupt Source Number		Vector	Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	Priority within IPR Setting Unit	Default Priority
NMI		11	H'0000002C to H'0000002F	16	—	—	High ↑
User break		12	H'00000030 to H'00000033	15	—	—	
IRQ	IRQ0	64	H'00000100 to H'00000103	0 to 15 (0)	IPR01 (15 to 12)	—	
	IRQ1	65	H'00000104 to H'00000107	0 to 15 (0)	IPR01 (11 to 8)	—	
	IRQ2	66	H'00000108 to H'0000010B	0 to 15 (0)	IPR01 (7 to 4)	—	
	IRQ3	67	H'0000010C to H'0000010F	0 to 15 (0)	IPR01 (3 to 0)	—	
	IRQ4	68	H'00000110 to H'00000113	0 to 15 (0)	IPR02 (15 to 12)	—	
	IRQ5	69	H'00000114 to H'00000117	0 to 15 (0)	IPR02 (11 to 8)	—	
	IRQ6	70	H'00000118 to H'0000011B	0 to 15 (0)	IPR02 (7 to 4)	—	
	IRQ7	71	H'0000011C to H'0000011F	0 to 15 (0)	IPR02 (3 to 0)	—	
RAM/ ROM	RAME	79	H'0000013C to H'0000013F	15	—	—	
	FIFE	82	H'00000148 to H'0000014B	15	—	—	
SINT	SINT15	93	H'00000174 to H'00000177	15	—	—	
	SINT14	94	H'00000178 to H'0000017B	14	—	—	
	SINT13	95	H'0000017C to H'0000017F	13	—	—	
	SINT12	96	H'00000180 to H'00000183	12	—	—	↓ Low

			Interrupt Vector					
Interrupt Source		Number	Vector	Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	Priority within IPR Setting Unit	Default Priority
SINT	SINT11		97	H'00000184 to H'00000187	11	—	—	High
	SINT10		98	H'00000188 to H'0000018B	10	—	—	
	SINT9		99	H'0000018C to H'0000018F	9	—	—	
	SINT8		100	H'00000190 to H'00000193	8	—	—	
	SINT7		101	H'00000194 to H'00000197	7	—	—	
	SINT6		102	H'00000198 to H'0000019B	6	—	—	
	SINT5		103	H'0000019C to H'0000019F	5	—	—	
	SINT4		104	H'000001A0 to H'000001A3	4	—	—	
	SINT3		105	H'000001A4 to H'000001A7	3	—	—	
	SINT2		106	H'000001A8 to H'000001AB	2	—	—	
	SINT1		107	H'000001AC to H'000001AF	1	—	—	
DMAC	DMAC0	DEI0	108	H'000001B0 to H'000001B3	0 to 15 (0)	IPR03 (15 to 12)	1	
		HEI0	109	H'000001B4 to H'000001B7			2	
	DMAC1	DEI1	112	H'000001C0 to H'000001C3	0 to 15 (0)	IPR03 (11 to 8)	1	
		HEI1	113	H'000001C4 to H'000001C7			2	
	DMAC2	DEI2	116	H'000001D0 to H'000001D3	0 to 15 (0)	IPR03 (7 to 4)	1	
		HEI2	117	H'000001D4 to H'000001D7			2	Low

			Interrupt Vector					
Interrupt Source		Number	Vector	Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	Priority within IPR Setting Unit	Default Priority
DMAC	DMAC3	DEI3	120	H'000001E0 to H'000001E3	0 to 15 (0)	IPR03 (3 to 0)	1	High
		HEI3	121	H'000001E4 to H'000001E7			2	
	DMAC4	DEI4	124	H'000001F0 to H'000001F3	0 to 15 (0)	IPR04 (15 to 12)	1	
		HEI4	125	H'000001F4 to H'000001F7			2	
	DMAC5	DEI5	128	H'00000200 to H'00000203	0 to 15 (0)	IPR04 (11 to 8)	1	
		HEI5	129	H'00000204 to H'00000207			2	
	DMAC6	DEI6	132	H'00000210 to H'00000213	0 to 15 (0)	IPR04 (7 to 4)	1	
		HEI6	133	H'00000214 to H'00000217			2	
	DMAC7	DEI7	136	H'00000220 to H'00000223	0 to 15 (0)	IPR04 (3 to 0)	1	
		HEI7	137	H'00000224 to H'00000227			2	
CMT	CMI0		140	H'00000230 to H'00000233	0 to 15 (0)	IPR05 (15 to 12)	—	
	CMI1		144	H'00000240 to H'00000243	0 to 15 (0)	IPR05 (11 to 8)	—	
WDT	ITI		148	H'00000250 to H'00000253	0 to 15 (0)	IPR05 (3 to 0)	—	
ATU-A	ICIA0		152	H'00000260 to H'00000263	0 to 15 (0)	IPR06 (15 to 12)	1	
	ICIA1		153	H'00000264 to H'00000267			2	
	ICIA2		156	H'00000270 to H'00000273	0 to 15 (0)	IPR06 (11 to 8)	1	
	ICIA3		157	H'00000274 to H'00000277			2	Low

			Interrupt Vector		Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	Priority within IPR Setting Unit	Default Priority
Interrupt Source Number			Vector	Vector Table Address Offset				
ATU-A	ICIA4		160	H'00000280 to H'00000283	0 to 15 (0)	IPR06 (7 to 4)	1	High
	ICIA5		161	H'00000284 to H'00000287			2	
	OVIA		164	H'00000290 to H'00000293	0 to 15 (0)	IPR06 (3 to 0)	—	
ATU-B	CMIB0		168	H'000002A0 to H'000002A3	0 to 15 (0)	IPR07 (15 to 12)	1	
	CMIB1		169	H'000002A4 to H'000002A7			2	
	CMIB6		172	H'000002B0 to H'000002B3	0 to 15 (0)	IPR07 (11 to 8)	1	
	ICIB0		173	H'000002B4 to H'000002B7			2	
ATU-C	ATU-C0	IMIC00	176	H'000002C0 to H'000002C3	0 to 15 (0)	IPR07 (7 to 4)	1	
		IMIC01	177	H'000002C4 to H'000002C7			2	
		IMIC02	178	H'000002C8 to H'000002CB			3	
		IMIC03	179	H'000002CC to H'000002CF			4	
		OVIC0	180	H'000002D0 to H'000002D3	0 to 15 (0)	IPR07 (3 to 0)	—	
	ATU-C1	IMIC10	184	H'000002E0 to H'000002E3	0 to 15 (0)	IPR08 (15 to 12)	1	
		IMIC11	185	H'000002E4 to H'000002E7			2	
		IMIC12	186	H'000002E8 to H'000002EB			3	
		IMIC13	187	H'000002EC to H'000002EF			4	
		OVIC1	188	H'000002F0 to H'000002F3	0 to 15 (0)	IPR08 (11 to 8)	—	Low

			Interrupt Vector					
Interrupt Source		Number	Vector	Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	Priority within IPR Setting Unit	Default Priority
ATU-C	ATU-C2	IMIC20	192	H'00000300 to H'00000303	0 to 15 (0)	IPR08 (7 to 4)	1	High
		IMIC21	193	H'00000304 to H'00000307			2	
		IMIC22	194	H'00000308 to H'0000030B			3	
		IMIC23	195	H'0000030C to H'0000030F			4	
		OVIC2	196	H'00000310 to H'00000313	0 to 15 (0)	IPR08 (3 to 0)	—	
	ATU-C3	IMIC30	200	H'00000320 to H'00000323	0 to 15 (0)	IPR09 (15 to 12)	1	
		IMIC31	201	H'00000324 to H'00000327			2	
		IMIC32	202	H'00000328 to H'0000032B			3	
		IMIC33	203	H'0000032C to H'0000032F			4	
		OVIC3	204	H'00000330 to H'00000333	0 to 15 (0)	IPR09 (11 to 8)	—	
	ATU-C4	IMIC40	208	H'00000340 to H'00000343	0 to 15 (0)	IPR09 (7 to 4)	1	
		IMIC41	209	H'00000344 to H'00000347			2	
		IMIC42	210	H'00000348 to H'0000034B			3	
		IMIC43	211	H'0000034C to H'0000034F			4	
		OVIC4	212	H'00000350 to H'00000353	0 to 15 (0)	IPR09 (3 to 0)	—	Low

			Interrupt Vector					
Interrupt Source		Number	Vector	Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	Priority within IPR Setting Unit	Default Priority
ATU-D	ATU-D0	CMID00	216	H'00000360 to H'00000363	0 to 15 (0)	IPR10 (15 to 12)	1	High
		CMID01	217	H'00000364 to H'00000367			2	
		CMID02	218	H'00000368 to H'0000036B			3	
		CMID03	219	H'0000036C to H'0000036F			4	
		OVI1D0	220	H'00000370 to H'00000373	0 to 15 (0)	IPR10 (11 to 8)	1	
		OVI2D0	221	H'00000374 to H'00000377			2	
		UDID00	224	H'00000380 to H'00000383	0 to 15 (0)	IPR10 (7 to 4)	1	
		UDID01	225	H'00000384 to H'00000387			2	
		UDID02	226	H'00000388 to H'0000038B			3	
		UDID03	227	H'0000038C to H'0000038F			4	
	ATU-D1	CMID10	228	H'00000390 to H'00000393	0 to 15 (0)	IPR10 (3 to 0)	1	
		CMID11	229	H'00000394 to H'00000397			2	
		CMID12	230	H'00000398 to H'0000039B			3	
		CMID13	231	H'0000039C to H'0000039F			4	
		OVI1D1	232	H'000003A0 to H'000003A3	0 to 15 (0)	IPR11 (15 to 12)	1	
		OVI2D1	233	H'000003A4 to H'000003A7			2	Low

Interrupt Source		Number	Interrupt Vector		Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	Priority within IPR Setting Unit	Default Priority
			Vector	Vector Table Address Offset				
ATU-D	ATU-D1	UDID10	236	H'000003B0 to H'000003B3	0 to 15 (0)	IPR11 (11 to 8)	1	High ↑
		UDID11	237	H'000003B4 to H'000003B7			2	
		UDID12	238	H'000003B8 to H'000003BB			3	
		UDID13	239	H'000003BC to H'000003BF			4	
	ATU-D2	CMID20	240	H'000003C0 to H'000003C3	0 to 15 (0)	IPR11 (7 to 4)	1	
		CMID21	241	H'000003C4 to H'000003C7			2	
		CMID22	242	H'000003C8 to H'000003CB			3	
		CMID23	243	H'000003CC to H'000003CF			4	
		OVI1D2	244	H'000003D0 to H'000003D3	0 to 15 (0)	IPR11 (3 to 0)	1	
		OVI2D2	245	H'000003D4 to H'000003D7			2	
		UDID20	248	H'000003E0 to H'000003E3	0 to 15 (0)	IPR12 (15 to 12)	1	
		UDID21	249	H'000003E4 to H'000003E7			2	
		UDID22	250	H'000003E8 to H'000003EB			3	
		UDID23	251	H'000003EC to H'000003EF			4	↓ Low

			Interrupt Vector					
Interrupt Source		**Number**	**Vector**	**Vector Table Address Offset**	**Interrupt Priority (Initial Value)**	**Corresponding IPR (Bit)**	**Priority within IPR Setting Unit**	**Default Priority**
ATU-D	ATU-D3	CMID30	252	H'000003F0 to H'000003F3	0 to 15 (0)	IPR12 (11 to 8)	1	High
		CMID31	253	H'000003F4 to H'000003F7			2	
		CMID32	254	H'000003F8 to H'000003FB			3	
		CMID33	255	H'000003FC to H'000003FF			4	
		OVI1D3	256	H'00000400 to H'00000403	0 to 15 (0)	IPR12 (7 to 4)	1	
		OVI2D3	257	H'00000404 to H'00000407			2	
		UDID30	260	H'00000410 to H'00000413	0 to 15 (0)	IPR12 (3 to 0)	1	
		UDID31	261	H'00000414 to H'00000417			2	
		UDID32	262	H'00000418 to H'0000041B			3	
		UDID33	263	H'0000041C to H'0000041F			4	
ATU-E	ATU-E0	CMIE00	288	H'00000480 to H'00000483	0 to 15 (0)	IPR14 (7 to 4)	1	
		CMIE01	289	H'00000484 to H'00000487			2	
		CMIE02	290	H'00000488 to H'0000048B			3	
		CMIE03	291	H'0000048C to H'0000048F			4	Low

			Interrupt Vector					
Interrupt Source		Number	Vector	Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	Priority within IPR Setting Unit	Default Priority
ATU-E	ATU-E1	CMIE10	292	H'00000490 to H'00000493	0 to 15 (0)	IPR14 (3 to 0)	1	High
		CMIE11	293	H'00000494 to H'00000497			2	
		CMIE12	294	H'00000498 to H'0000049B			3	
		CMIE13	295	H'0000049C to H'0000049F			4	
	ATU-E2	CMIE20	296	H'000004A0 to H'000004A3	0 to 15 (0)	IPR15 (15 to 12)	1	
		CMIE21	297	H'000004A4 to H'000004A7			2	
		CMIE22	298	H'000004A8 to H'000004AB			3	
		CMIE23	299	H'000004AC to H'000004AF			4	
	ATU-E3	CMIE30	300	H'000004B0 to H'000004B3	0 to 15 (0)	IPR15 (11 to 8)	1	
		CMIE31	301	H'000004B4 to H'000004B7			2	
		CMIE32	302	H'000004B8 to H'000004BB			3	
		CMIE33	303	H'000004BC to H'000004BF			4	
	ATU-E4	CMIE40	304	H'000004C0 to H'000004C3	0 to 15 (0)	IPR15 (7 to 4)	1	
		CMIE41	305	H'000004C4 to H'000004C7			2	
		CMIE42	306	H'000004C8 to H'000004CB			3	
		CMIE43	307	H'000004CC to H'000004CF			4	Low

Interrupt Source		Number	Interrupt Vector		Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	Priority within IPR Setting Unit	Default Priority
			Vector	Vector Table Address Offset				
ATU-E	ATU-E5	CMIE50	308	H'000004D0 to H'000004D3	0 to 15 (0)	IPR15 (3 to 0)	1	High ↑
		CMIE51	309	H'000004D4 to H'000004D7			2	
		CMIE52	310	H'000004D8 to H'000004DB			3	
		CMIE53	311	H'000004DC to H'000004DF			4	
ATU-F	ICIF0		312	H'000004E0 to H'000004E3	0 to 15 (0)	IPR16 (15 to 12)	1	
	ICIF1		313	H'000004E4 to H'000004E7			2	
	ICIF2		314	H'000004E8 to H'000004EB			3	
	ICIF3		315	H'000004EC to H'000004EF			4	
	ICIF4		316	H'000004F0 to H'000004F3	0 to 15 (0)	IPR16 (11 to 8)	1	
	ICIF5		317	H'000004F4 to H'000004F7			2	
	ICIF6		318	H'000004F8 to H'000004FB			3	
	ICIF7		319	H'000004FC to H'000004FF			4	
	ICIF8		320	H'00000500 to H'00000503	0 to 15 (0)	IPR16 (7 to 4)	1	
	ICIF9		321	H'00000504 to H'00000507			2	
	ICIF10		322	H'00000508 to H'0000050B			3	
	ICIF11		323	H'0000050C to H'0000050F			4	↓ Low

		Interrupt Vector					
Interrupt Source Number		Vector	Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	Priority within IPR Setting Unit	Default Priority
ATU-F	ICIF12	324	H'00000510 to H'00000513	0 to 15 (0)	IPR16 (3 to 0)	1	High ↑
	ICIF13	325	H'00000514 to H'00000517			2	
	ICIF14	326	H'00000518 to H'0000051B			3	
	ICIF15	327	H'0000051C to H'0000051F			4	
	ICIF16	328	H'00000520 to H'00000523	0 to 15 (0)	IPR17 (15 to 12)	1	
	ICIF17	329	H'00000524 to H'00000527			2	
	ICIF18	330	H'00000528 to H'0000052B			3	
	ICIF19	331	H'0000052C to H'0000052F			4	
	ICIF20	332	H'00000530 to H'00000533		IPR17 (11 to 8)	1	
	ICIF21	333	H'00000534 to H'00000537			2	
	ICIF22	334	H'00000538 to H'0000053B			3	
	ICIF23	335	H'0000053C to H'0000053F			4	
	ICIF24	336	H'00000540 to H'00000543		IPR17 (7 to 4)	1	
	ICIF25	337	H'00000544 to H'00000547			2	
	ICIF26	338	H'00000548 to H'0000054B			3	
	ICIF27	339	H'0000054C to H'0000054F			4	↓ Low

		Interrupt Vector					
Interrupt Source Number		Vector	Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	Priority within IPR Setting Unit	Default Priority
ATU-F	OVIF0	340	H'00000550 to H'00000553	0 to 15 (0)	IPR18 (15 to 12)	1	High
	OVIF1	341	H'00000554 to H'00000557			2	
	OVIF2	342	H'00000558 to H'0000055B			3	
	OVIF3	343	H'0000055C to H'0000055F			4	
	OVIF4	344	H'00000560 to H'00000563	0 to 15 (0)	IPR18 (11 to 8)	1	
	OVIF5	345	H'00000564 to H'00000567			2	
	OVIF6	346	H'00000568 to H'0000056B			3	
	OVIF7	347	H'0000056C to H'0000056F			4	
	OVIF8	348	H'00000570 to H'00000573	0 to 15 (0)	IPR18 (7 to 4)	1	
	OVIF9	349	H'00000574 to H'00000577			2	
	OVIF10	350	H'00000578 to H'0000057B			3	
	OVIF11	351	H'0000057C to H'0000057F			4	
	OVIF12	352	H'00000580 to H'00000583	0 to 15 (0)	IPR18 (3 to 0)	1	
	OVIF13	353	H'00000584 to H'00000587			2	
	OVIF14	354	H'00000588 to H'0000058B			3	
	OVIF15	355	H'0000058C to H'0000058F			4	Low

		Interrupt Vector					
Interrupt Source Number		Vector	Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	Priority within IPR Setting Unit	Default Priority
ATU-F	OVIF16	356	H'00000590 to H'00000593	0 to 15 (0)	IPR19 (15 to 12)	1	High
	OVIF17	357	H'00000594 to H'00000597			2	
	OVIF18	358	H'00000598 to H'0000059B			3	
	OVIF19	359	H'0000059C to H'0000059F			4	
	OVIF20	360	H'000005A0 to H'000005A3		IPR19 (11 to 8)	1	
	OVIF21	361	H'000005A4 to H'000005A7			2	
	OVIF22	362	H'000005A8 to H'000005AB			3	
	OVIF23	363	H'000005AC to H'000005AF			4	
	OVIF24	364	H'000005B0 to H'000005B3		IPR19 (7 to 4)	1	
	OVIF25	365	H'000005B4 to H'000005B7			2	
	OVIF26	366	H'000005B8 to H'000005BB			3	
	OVIF27	367	H'000005BC to H'000005BF			4	
ATU-G	CMIG0	368	H'000005C0 to H'000005C3		IPR20 (15 to 12)	1	
	CMIG1	369	H'000005C4 to H'000005C7			2	
	CMIG2	370	H'000005C8 to H'000005CB			3	
	CMIG3	371	H'000005CC to H'000005CF			4	
	CMIG4	372	H'000005D0 to H'000005D3		IPR20 (11 to 8)	1	
	CMIG5	373	H'000005D4 to H'000005D7			2	Low

		Interrupt Vector					
Interrupt Source	Number	Vector	Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	Priority within IPR Setting Unit	Default Priority
ATU-H	CMIH	376	H'000005E0 to H'000005E3		IPR20 (7 to 4)	4	High
ATU-J	DFIJ0	380	H'000005F0 to H'000005F3	0 to 15 (0)	IPR21 (15 to 12)	1	
	DFIJ1	381	H'000005F4 to H'000005F7			2	
	OVIJ0	384	H'00000600 to H'00000603	0 to 15 (0)	IPR21 (11 to 8)	1	
	OVIJ1	385	H'00000604 to H'00000607			2	
	DOVIJ0	388	H'00000610 to H'00000613	0 to 15 (0)	IPR21 (7 to 4)	1	
	DOVIJ1	389	H'00000614 to H'00000617			2	
ADC	ADI0	392	H'00000620 to H'00000623	0 to 15 (0)	IPR22 (15 to 12)	—	
	ADI1	396	H'00000630 to H'00000633	0 to 15 (0)	IPR22 (11 to 8)	—	
	ADID0	400	H'00000640 to H'00000643	0 to 15 (0)	IPR22 (7 to 4)	1	
	ADID1	401	H'00000644 to H'00000647			2	
	ADID2	402	H'00000648 to H'0000064B			3	
	ADID3	403	H'0000064C to H'0000064F			4	Low

		Interrupt Vector					
Interrupt Source Number		Vector	Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	Priority within IPR Setting Unit	Default Priority
ADC	ADID4	404	H'00000650 to H'00000653	0 to 15 (0)	IPR22 (3 to 0)	1	High
	ADID5	405	H'00000654 to H'00000657			2	
	ADID6	406	H'00000658 to H'0000065B			3	
	ADID7	407	H'0000065C to H'0000065F			4	
	ADID8	408	H'00000660 to H'00000663	0 to 15 (0)	IPR23 (15 to 12)	1	
	ADID9	409	H'00000664 to H'00000667			2	
	ADID10	410	H'00000668 to H'0000066B			3	
	ADID11	411	H'0000066C to H'0000066F			4	
	ADID12	412	H'00000670 to H'00000673	0 to 15 (0)	IPR23 (11 to 8)	1	
	ADID13	413	H'00000674 to H'00000677			2	
	ADID14	414	H'00000678 to H'0000067B			3	
	ADID15	415	H'0000067C to H'0000067F			4	
	ADID40	416	H'00000680 to H'00000683	0 to 15 (0)	IPR23 (7 to 4)	—	
	ADID41	417	H'00000684 to H'00000687	0 to 15 (0)	IPR23 (3 to 0)	—	
	ADID42	418	H'00000688 to H'0000068B	0 to 15 (0)	IPR24 (15 to 12)	—	
	ADID43	419	H'0000068C to H'0000068F	0 to 15 (0)	IPR24 (11 to 8)	—	
	ADID44	420	H'00000690 to H'00000693	0 to 15 (0)	IPR24 (7 to 4)	—	Low

			Interrupt Vector					
Interrupt Source Number			Vector	Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	Priority within IPR Setting Unit	Default Priority
ADC	ADID45		421	H'00000694 to H'00000697	0 to 15 (0)	IPR24 (3 to 0)	—	High
	ADID46		422	H'00000698 to H'0000069B	0 to 15 (0)	IPR25 (15 to 12)	—	
	ADID47		423	H'0000069C to H'0000069F	0 to 15 (0)	IPR25 (11 to 8)	—	
SCI	SCI_A	ERIA	424	H'000006A0 to H'000006A3	0 to 15 (0)	IPR26 (15 to 12)	1	
		RXIA	425	H'000006A4 to H'000006A7			2	
		TXIA	426	H'000006A8 to H'000006AB			3	
		TEIA	427	H'000006AC to H'000006AF			4	
	SCI_B	ERIB	428	H'000006B0 to H'000006B3	0 to 15 (0)	IPR26 (11 to 8)	1	
		RXIB	429	H'000006B4 to H'000006B7			2	
		TXIB	430	H'000006B8 to H'000006BB			3	
		TEIB	431	H'000006BC to H'000006BF			4	
	SCI_C	ERIC	432	H'000006C0 to H'000006C3	0 to 15 (0)	IPR26 (7 to 4)	1	
		RXIC	433	H'000006C4 to H'000006C7			2	
		TXIC	434	H'000006C8 to H'000006CB			3	
		TEIC	435	H'000006CC to H'000006CF			4	Low

			Interrupt Vector		Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	Priority within IPR Setting Unit	Default Priority
Interrupt Source		Number	Vector	Vector Table Address Offset				
SCI	SCI_D	ERID	436	H'000006D0 to H'000006D3	0 to 15 (0)	IPR26 (3 to 0)	1	High
		RXID	437	H'000006D4 to H'000006D7			2	
		TXID	438	H'000006D8 to H'000006DB			3	
		TEID	439	H'000006DC to H'000006DF			4	
	SCI_E	ERIE	440	H'000006E0 to H'000006E3	0 to 15 (0)	IPR27 (15 to 12)	1	
		RXIE	441	H'000006E4 to H'000006E7			2	
		TXIE	442	H'000006E8 to H'000006EB			3	
		TEIE	443	H'000006EC to H'000006EF			4	
RSPI	RSPI_A	SPEIA	444	H'000006F0 to H'000006F3	0 to 15 (0)	IPR27 (11 to 8)	1	
		SPRIA	445	H'000006F4 to H'000006F7			2	
		SPTIA	446	H'000006F8 to H'000006FB			3	
	RSPI_B	SPEIB	448	H'00000700 to H'00000703	0 to 15 (0)	IPR27 (7 to 4)	1	
		SPRIB	449	H'00000704 to H'00000707			2	
		SPTIB	450	H'00000708 to H'0000070B			3	
	RSPI_C	SPEIC	452	H'00000710 to H'00000713	0 to 15 (0)	IPR27 (3 to 0)	1	
		SPRIC	453	H'00000714 to H'00000717			2	
		SPTIC	454	H'00000718 to H'0000071B			3	Low

			Interrupt Vector					
Interrupt Source		Number	Vector	Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	Priority within IPR Setting Unit	Default Priority
RCAN -TL1	RCAN_A	ERSA	456	H'00000720 to H'00000723	0 to 15 (0)	IPR28 (15 to 12)	1	High
		OVRA	457	H'00000724 to H'00000727			2	
		RMA0	458	H'00000728 to H'0000072B			3	
		RMA1	459	H'0000072C to H'0000072F			4	
		SLEA	460	H'00000730 to H'00000733			5	
		MBEA	461	H'00000734 to H'00000737			6	
	RCAN_B	ERSB	464	H'00000740 to H'00000743	0 to 15 (0)	IPR28 (11 to 8)	1	
		OVRB	465	H'00000744 to H'00000747			2	
		RMB0	466	H'00000748 to H'0000074B			3	
		RMB1	467	H'0000074C to H'0000074F			4	
		SLEB	468	H'00000750 to H'00000753			5	
		MBEB	469	H'00000754 to H'00000757			6	
	RCAN_C	ERSC	472	H'00000760 to H'00000763	0 to 15 (0)	IPR28 (7 to 4)	1	
		OVRC	473	H'00000764 to H'00000767			2	
		RMC0	474	H'00000768 to H'0000076B			3	
		RMC1	475	H'0000076C to H'0000076F			4	
		SLEC	476	H'00000770 to H'00000773			5	
		MBEC	477	H'00000774 to H'00000777			6	Low

			Interrupt Vector					
Interrupt Source		Number	Vector	Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	Priority within IPR Setting Unit	Default Priority
RCAN-TL1	RCAN_D	ERSD	480	H'00000780 to H'00000783	0 to 15 (0)	IPR28 (3 to 0)	1	High
		OVRD	481	H'00000784 to H'00000787			2	
		RMD0	482	H'00000788 to H'0000078B			3	
		RMD1	483	H'0000078C to H'0000078F			4	
		SLED	484	H'00000790 to H'00000793			5	
		MBED	485	H'00000794 to H'00000797			6	
A-DMAC	TE74		488	H'000007A0 to H'000007A3		IPR29 (15 to 12)	—	
	TE75		489	H'000007A4 to H'000007A7		IPR29 (11 to 8)	—	
ATU-E	ATU-E6	CMIE60	492	H'000007B0 to H'000007B3		IPR21 (3 to 0)	1	
		CMIE61	493	H'000007B4 to H'000007B7			2	
		CMIE62	494	H'000007B8 to H'000007BB			3	
		CMIE63	495	H'000007BC to H'000007BF			4	
Flex Ray		FR0	496	H'000007C0 to H'000007C3		IPR30 (15 to 12)	—	
		FR1	497	H'000007C4 to H'000007C7		IPR30 (11 to 8)	—	
		FRT0	498	H'000007C8 to H'000007CB		IPR30 (7 to 4)	—	
		FRT1	499	H'000007CC to H'000007CF		IPR30 (3 to 0)	—	Low

8.6 Operation

8.6.1 Interrupt Operation Sequence

The sequence of interrupt operations is described below. Figure 8.2 shows the operation flow.

1. The interrupt request sources requests an interrupt to the interrupt controller.
2. The interrupt controller selects the highest-priority interrupt from the received interrupts according to the priority levels set in interrupt priority registers 01 to 30 (IPR01 to IPR30). Remaining interrupts are ignored*. If multiple interrupts have the same IPR priority level or if multiple interrupts occur within a single module, the interrupt with the highest priority is selected according to the priority within the IPR setting unit and default priority shown in table 8.4.
3. The priority level of the interrupt selected by the interrupt controller is compared with the interrupt level mask bits (I3 to I0) in the status register (SR) of the CPU. If the interrupt request priority level is equal to or less than the level set in bits I3 to I0, the interrupt request is ignored. If the interrupt request priority level is higher than the level in bits I3 to I0, the interrupt controller receives the interrupt and requests an interrupt to the CPU.
4. The CPU detects the interrupt requested by the interrupt controller when the CPU decodes the instruction to be executed. Instead of executing the decoded instruction, the CPU starts interrupt exception handling (figure 8.4).
5. The start address of the interrupt exception handler corresponding to the received interrupt is fetched from the exception handling vector table.
6. The contents of the status register (SR) are saved onto the stack, and the priority level of the received interrupt is copied to bits I3 to I0 in SR.
7. The contents of the program counter (PC) are saved onto the stack.
8. The CPU branches to the fetched start address of the interrupt exception handler and starts executing the program. The branch is not delayed branch.

Notes: The interrupt source flag should be cleared in the interrupt handler. The time from when the interrupt source flag is cleared to when the interrupt signal is negated in the CPU is the same time which is described as the time from occurrence of interrupt request until interrupt controller identifies priority in table 8.5. Therefore, read the interrupt source flag after clearing it so that an interrupt request that should have been cleared is not received again erroneously. After that, execute the RTE instruction.

* Interrupt requests that are designated as edge-sensing are held pending until the interrupt requests are received. IRQ interrupts, however, can be cancelled by accessing the IRQ interrupt request register (IRQRR). For details, see section 8.4.3, IRQ Interrupts.
Interrupts held pending due to edge-sensing are cleared by a power-on reset.

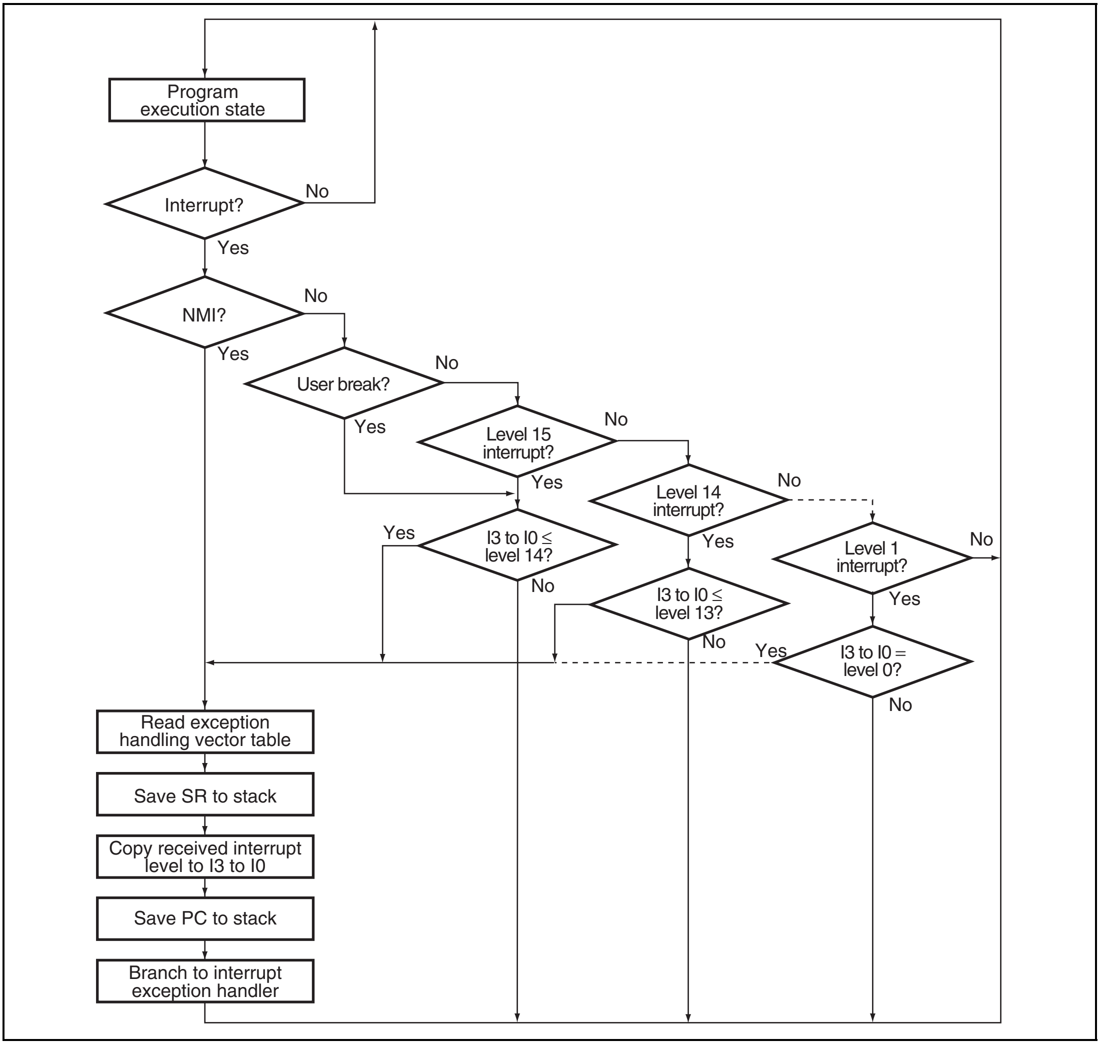
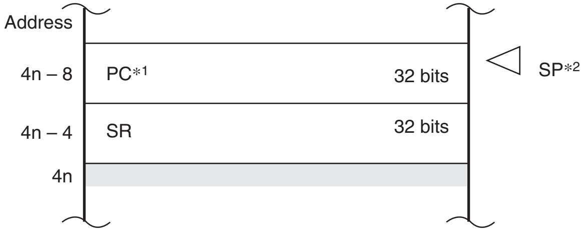


Figure 8.2 Interrupt Operation Flow

8.6.2 Stack after Interrupt Exception Handling

Figure 8.3 shows the stack after interrupt exception handling.



Notes: 1. PC: Start address of the next instruction (instruction at return address) after the executed instruction
2. Always make sure that SP is a multiple of 4.

Figure 8.3 Stack after Interrupt Exception Handling

8.7 Interrupt Response Time

Table 8.5 lists the interrupt response time, which is the time from the occurrence of an interrupt request until the interrupt exception handling starts and fetching of the first instruction in the exception handler begins. The interrupt processing operations differ in the cases when banking is disabled, when banking is enabled without register bank overflow, and when banking is enabled with register bank overflow. Figures 8.4 and 8.5 show examples of pipeline operation when banking is disabled. Figures 8.6 and 8.7 show examples of pipeline operation when banking is enabled without register bank overflow. Figures 8.8 and 8.9 show examples of pipeline operation when banking is enabled with register bank overflow.

Note that in table 8.5, and figures 8.4 to 8.9, Icyc represents ϕ cycle, and Bcyc and Pcyc represent Pϕ cycle.

Table 8.5 Interrupt Response Time

			Number of Cycles					
Item			NMI	User Break	IRQ	SINT	Peripheral Module	Remarks
Time from occurrence of interrupt request until interrupt controller identifies priority, compares it with mask bits in SR, and sends interrupt request signal to CPU			2 Icyc + 3 Pcyc	3 Icyc	2 Icyc + 1 Bcyc + 3 Pcyc	2 Icyc + 1 Bcyc	2 Icyc + 1 Bcyc + 1 Pcyc	
Time from input of interrupt request signal to CPU until sequence currently being executed is completed, interrupt exception handling starts, and first instruction in interrupt exception handler is fetched	No register banking	Min.	3 Icyc + m1 + m2					Min. is when the interrupt wait time is zero. Max. is when a higher-priority interrupt request has occurred during interrupt exception handling.
		Max.	4 Icyc + 2(m1 + m2) + m3					
	Register banking without register bank overflow	Min.	—	—	3 Icyc + m1 + m2			Min. is when the interrupt wait time is zero. Max. is when an interrupt request has occurred during execution of the RESBANK instruction.
		Max.	—	—	12 Icyc + m1 + m2			
	Register banking with register bank overflow	Min.	—	—	3 Icyc + m1 + m2			Min. is when the interrupt wait time is zero. Max. is when an interrupt request has occurred during execution of the RESBANK instruction.
		Max.	—	—	3 Icyc + m1 + m2 + 19(m4)			

			Number of States					
Item			**NMI**	**User Break**	**IRQ**	**SINT**	**Peripheral Module**	**Remarks**
Interrupt response time	No register banking	Min.	5 Icyc + 3 Pcyc + m1 + m2	6 Icyc + m1 + m2	5 Icyc + 1 Bcyc + 3 Pcyc + m1 + m2	5 Icyc + 1 Bcyc + m1 + m2	5 Icyc + 1 Bcyc + 1 Pcyc + m1 + m2	200-MHz operation*1*2: 0.040 to 0.135 µs
		Max.	6 Icyc + 3 Pcyc + 2 (m1 + m2) + m3	7 Icyc + 2 (m1 + m2) + m3	6 Icyc + 1 Bcyc + 3 Pcyc + 2 (m1 + m2) + m3	6 Icyc + 1 Bcyc + 2 (m1 + m2) + m3	6 Icyc + 1 Bcyc + 1 Pcyc + 2 (m1 + m2) + m3	200-MHz operation*1*2: 0.060 to 0.155 µs
	Register banking without register bank overflow	Min.	—	—	5 Icyc + 1 Bcyc + 3 Pcyc + m1 + m2	5 Icyc + 1 Bcyc + m1 + m2	5 Icyc + 1 Bcyc + 1 Pcyc + m1 + m2	200-MHz operation*1*2: 0.060 to 0.135 µs
		Max.	—	—	14 Icyc + 1 Bcyc + 3 Pcyc + m1 + m2	14 Icyc + 1 Bcyc + m1 + m2	14 Icyc + 1 Bcyc + 1 Pcyc + m1 + m2	200-MHz operation*1*2: 0.105 to 0.180 µs
	Register banking with register bank overflow	Min.	—	—	5 Icyc + 1 Bcyc + 3 Pcyc + m1 + m2	5 Icyc + 1 Bcyc + m1 + m2	5 Icyc + 1 Bcyc + 1 Pcyc + m1 + m2	200-MHz operation*1*2: 0.060 to 0.135 µs
		Max.	—	—	5 Icyc + 1 Bcyc + 3 Pcyc + m1 + m2 + 19 (m4)	5 Icyc + 1 Bcyc + m1 + m2 + 19 (m4)	5 Icyc + 1 Bcyc + 1 Pcyc + m1 +m2 + 19 (m4)	200-MHz operation*1*2: 0.155 to 0.230 µs

Notes: m1 to m4 denotes the number of states needed for the following memory accesses.

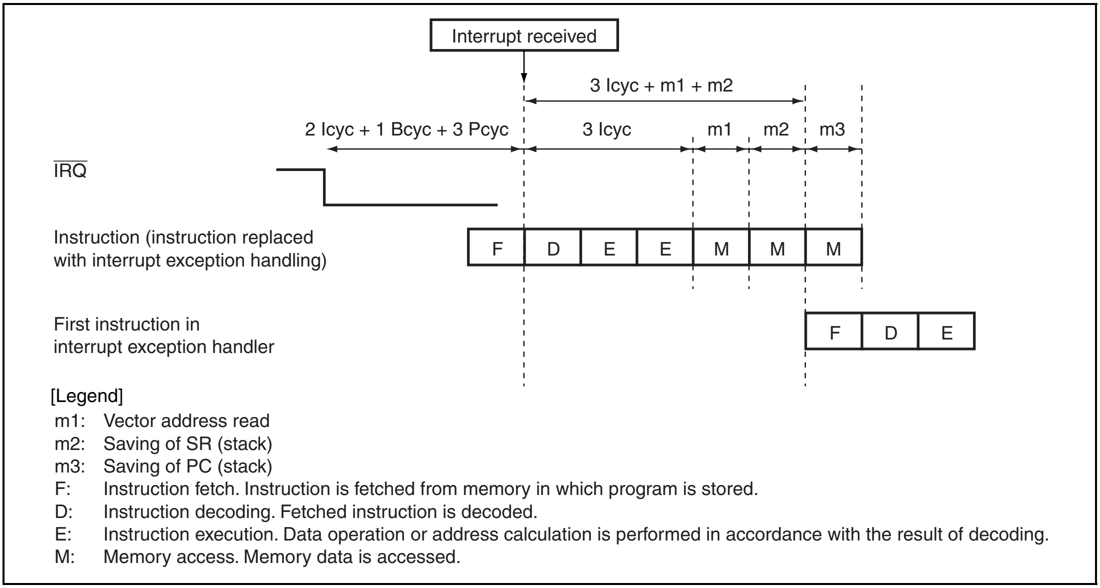
m1: Vector address read (longword read)

m2: SR saving (longword write)

m3: PC saving (longword write)

m4: Restoring banked registers (R0 to R14, GBR, MACH, MACL, and PR) from the stack.

1. When m1 = m2 = m3 = m4 = 1 Icyc
2. When ϕ and Pϕ are 200 and 40 MHz, respectively


Interrupt received
3 Icyc + m1 + m2
2 Icyc + 1 Bcyc + 3 Pcyc
3 Icyc
m1
m2
m3
IRQ
Instruction (instruction replaced with interrupt exception handling)
F
D
E
E
M
M
M
First instruction in interrupt exception handler
F
D
E

[Legend]

m1: Vector address read

m2: Saving of SR (stack)

m3: Saving of PC (stack)

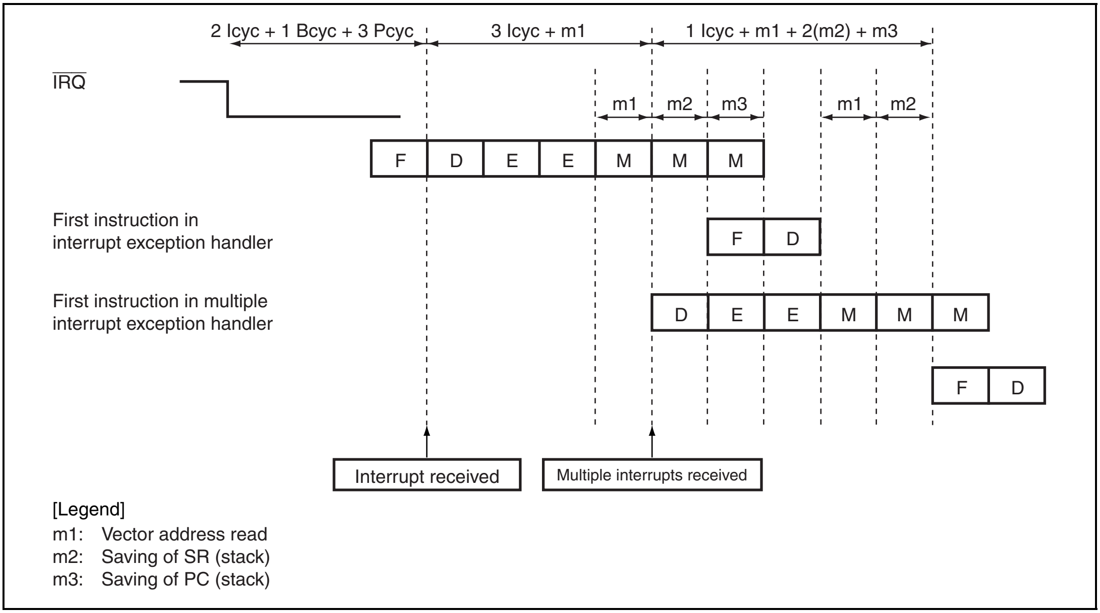
F: Instruction fetch. Instruction is fetched from memory in which program is stored.

D: Instruction decoding. Fetched instruction is decoded.

E: Instruction execution. Data operation or address calculation is performed in accordance with the result of decoding.

M: Memory access. Memory data is accessed.

Figure 8.4 Example of Pipeline Operation when IRQ Interrupt is Received (No Register Banking)


2 Icyc + 1 Bcyc + 3 Pcyc
3 Icyc + m1
1 Icyc + m1 + 2(m2) + m3
IRQ
m1
m2
m3
m1
m2
F
D
E
E
M
M
M
First instruction in interrupt exception handler
F
D
First instruction in multiple interrupt exception handler
D
E
E
M
M
M
F
D
Interrupt received
Multiple interrupts received

[Legend]

m1: Vector address read

m2: Saving of SR (stack)

m3: Saving of PC (stack)

Figure 8.5 Example of Pipeline Operation for Multiple Interrupts (No Register Banking)

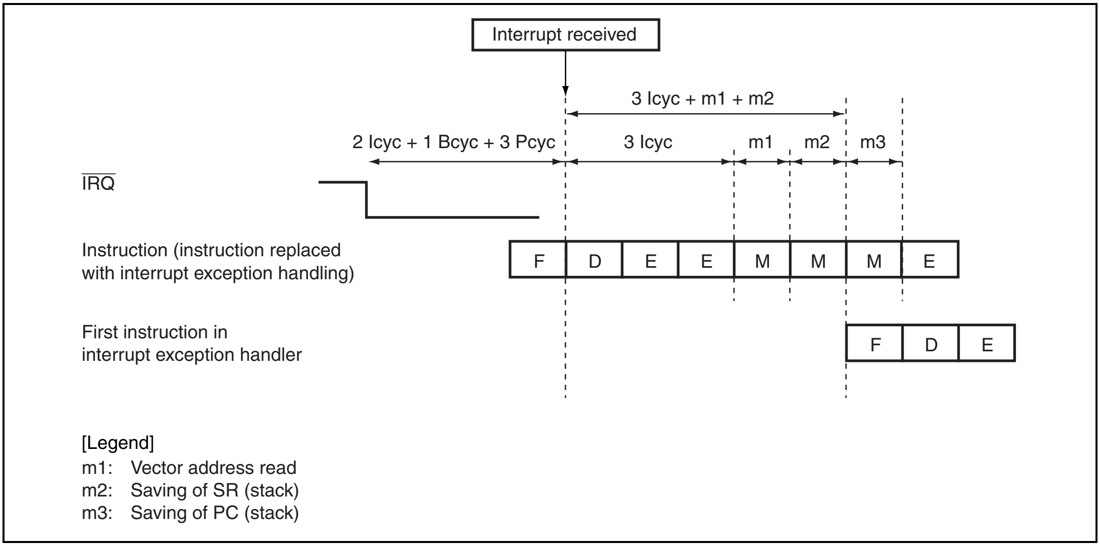


Figure 8.6 Example of Pipeline Operation when IRQ Interrupt is Received (Register Banking without Register Bank Overflow)

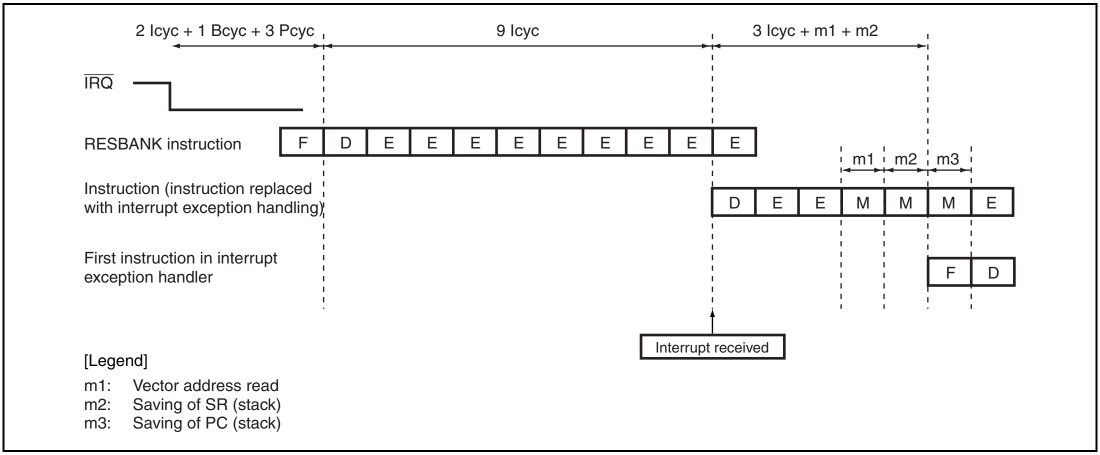


Figure 8.7 Example of Pipeline Operation when Interrupt is Received during RESBANK Instruction Execution (Register Banking without Register Bank Overflow)

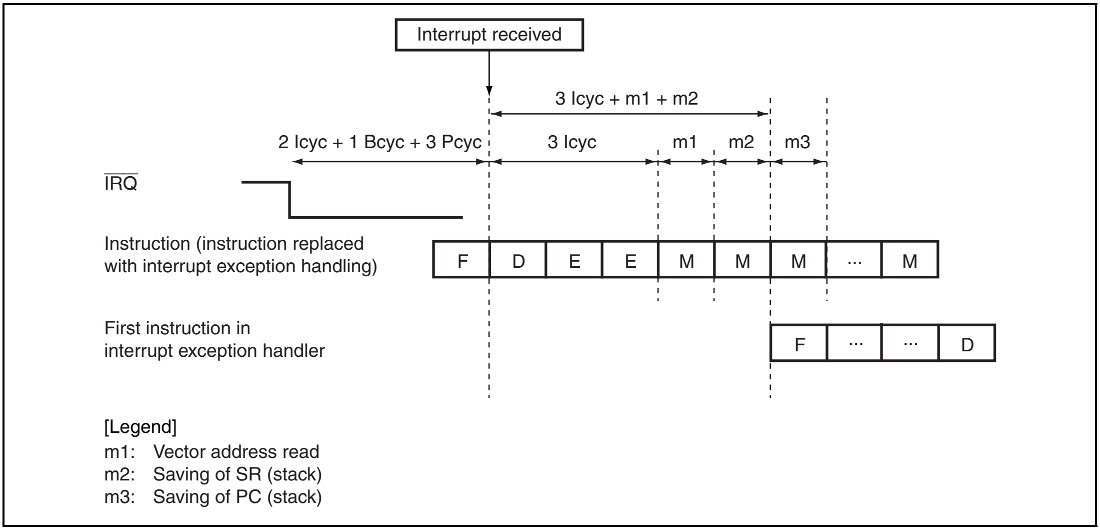


Figure 8.8 Example of Pipeline Operation when IRQ Interrupt is Received (Register Banking with Register Bank Overflow)

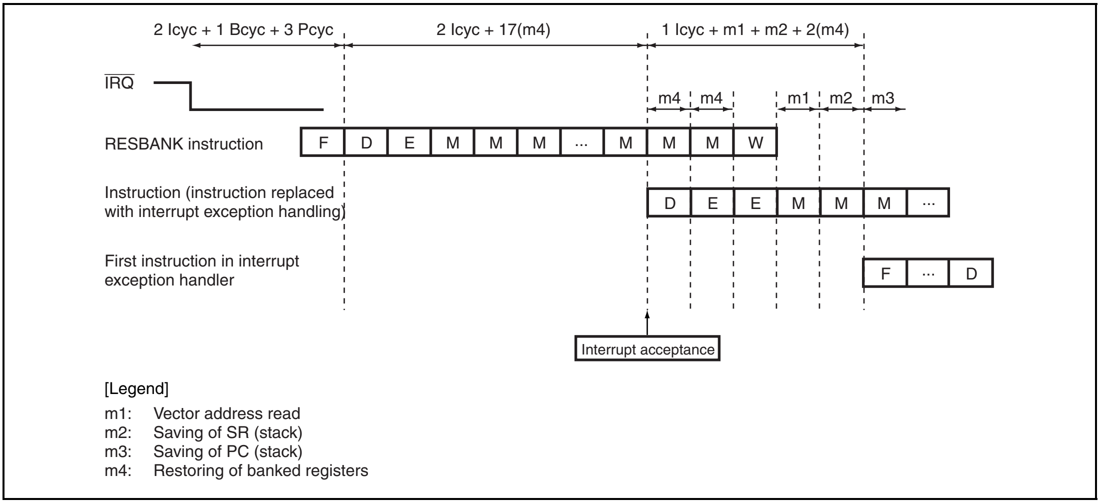


Figure 8.9 Example of Pipeline Operation when Interrupt is Received during RESBANK Instruction Execution (Register Banking with Register Bank Overflow)

8.8 Register Banks

This LSI has fifteen register banks used to save and restore registers for the interrupt processing at high speed. Figure 8.10 is the register bank configuration.

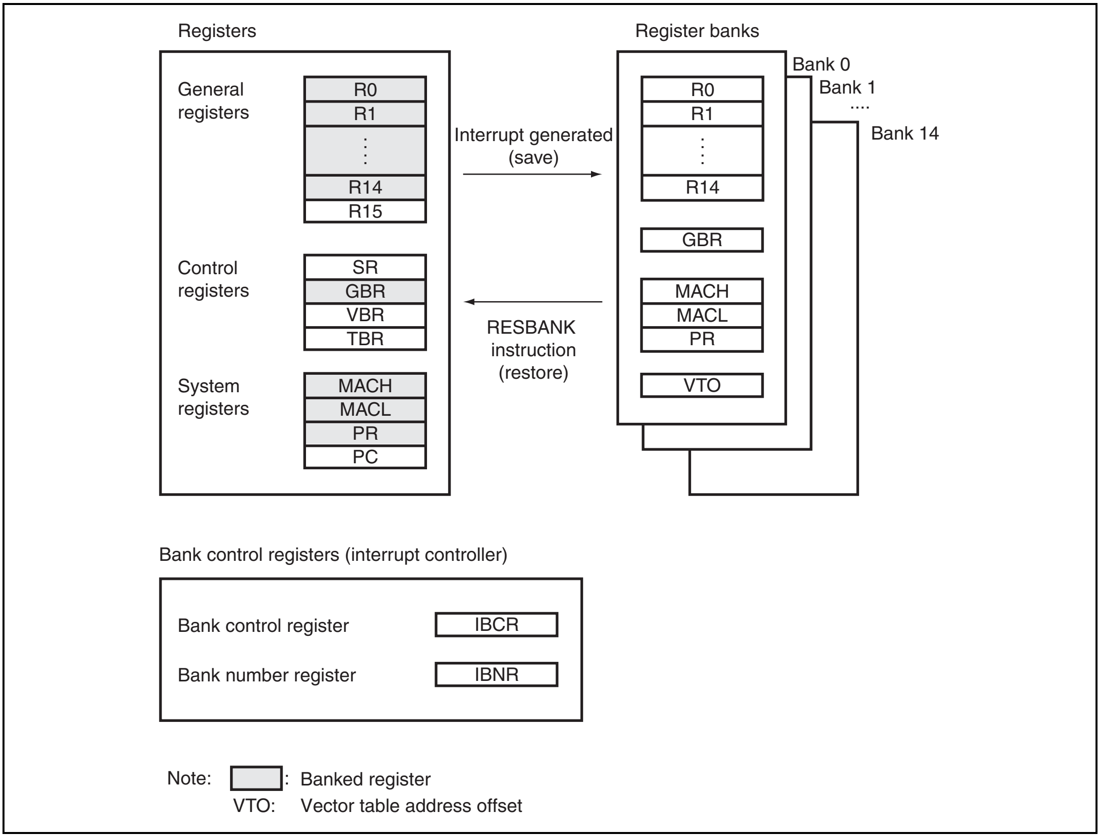


Figure 8.10 Overview of Register Bank Configuration

8.8.1 Banked Registers and Input/Output Method

(1) Banked Registers

The contents of the general registers (R0 to R14), global base register (GBR), multiply and accumulate registers (MACH and MACL), and procedure register (PR), and the vector table address offset are banked.

(2) Input/Output Configuration

This LSI has fifteen register banks, bank 0 to bank 14. Register banks are stacked in first-in last-out (FILO) sequence. Saving takes place in the order of the bank number from 0 to 14 and restoring takes place in the reverse order from the last bank saved.

8.8.2 Bank Saving and Restoring Operations

(1) Saving to Bank

Figure 8.11 shows register bank saving operations. The following operations are performed when an interrupt for which usage of register banks is allowed is received by the CPU:

(a) Assume that the bank number bit (BN) in the bank number register (IBNR) before the interrupt is generated is i.

(b) The contents of registers R0 to R14, GBR, MACH, MACL, and PR, and the vector table address offset (VTO) of the received interrupt are saved in bank i.

(c) The BN bit is incremented by 1.

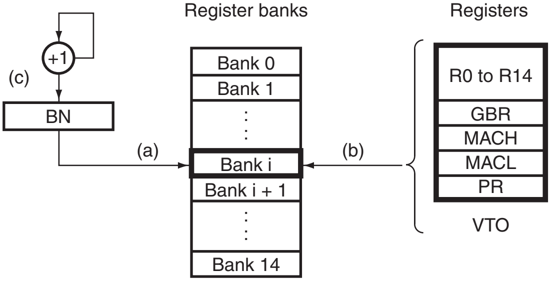


Figure 8.11 Bank Saving Operation

Figure 8.12 shows the timing for saving registers to a register bank. Saving registers to a register bank takes place between the start of interrupt exception handling and the start of fetching the first instruction in the interrupt exception handler.

In figure 8.12, Icyc represents the cycle of the ϕ clock and Bcyc and Pcyc represent the cycle of the Pϕ clock.

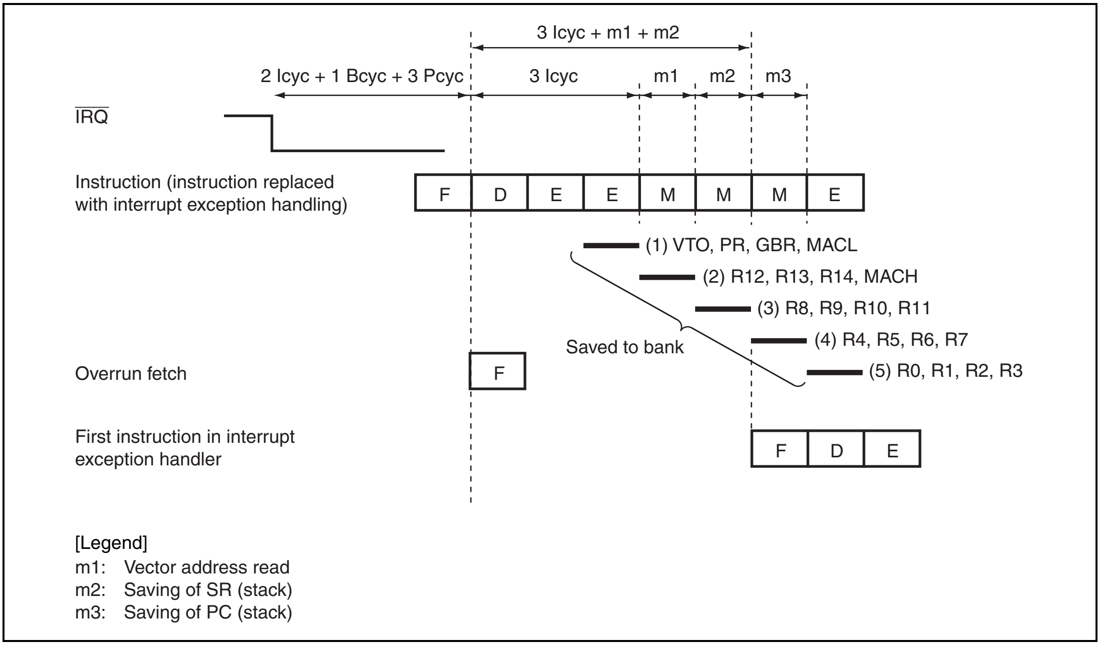


Figure 8.12 Bank Save Timing

(2) Restoring from Bank

The RESBANK (restoring from register bank) instruction is used to restore data saved in a register bank. After restoring data from the register banks with the RESBANK instruction at the end of the interrupt exception handler, execute the RTE instruction to return from the interrupt exception handler.

8.8.3 Saving and Restoring Operations after Saving Registers to All Banks

Assume that all register banks has been used for saving registers when an interrupt for which usage of the register banks is enabled is received by the CPU. When the BOVE bit in the bank number register (IBNR) is cleared to 0, registers are automatically saved to the stack area instead of saving to a register bank. When the BOVE bit in IBNR is set to 1, a register bank overflow exception occurs and registers are not saved to the stack area.

Saving and restoring operations when using the stack are shown below.

(1) Saving to Stack

1. The status register (SR) and program counter (PC) are saved to the stack during interrupt exception handling.
2. The contents of the banked registers (R0 to R14, GBR, MACH, MACL, and PR) are saved to the stack. The registers are saved to the stack in the order of MACL, MACH, GBR, PR, R14, R13, …, R1, and R0.
3. The register bank overflow bit (BO) in SR is set to 1.
4. The bank number bit (BN) value in the bank number register (IBNR) remains set to a maximum value of 15.

(2) Restoring from Stack

Operations when the RESBANK (restoring from register bank) instruction is executed with the register bank overflow bit (BO) in SR set to 1 are shown below.

1. The contents of the banked registers (R0 to R14, GBR, MACH, MACL, and PR) are restored from the stack. The registers are restored from the stack in the order of R0, R1, …, R13, R14, PR, GBR, MACH, and MACL.
2. The bank number bit (BN) value in the bank number register (IBNR) remains set to a maximum value of 15.

8.8.4 Register Bank Exception

There are two register bank exceptions (register bank errors): a register bank overflow and a register bank underflow.

(1) Register Bank Overflow

Assume that all register banks has been used for saving registers when an interrupt for which usage of the register banks is enabled is received by the CPU. When the BOVE bit in IBNR is set to 1, a register bank overflow exception occurs and registers are not saved to the stack area.

(2) Register Bank Underflow

This exception occurs if the RESBANK (restoring from register bank) instruction is executed when no data has been saved to the register banks. In this case, the contents of R0 to R14, GBR, MACH, MACL, and PR do not change. In addition, the bank number bit (BN) value in the bank number register (IBNR) remains set to 0.

8.8.5 Register Bank Error Exception Handling

When a register bank error occurs, register bank error exception handling starts. When this happens, the CPU operation is shown below.

1. The start address of the exception handler for the register bank error is fetched from the exception handling vector table.
2. The status register (SR) is saved to the stack.
3. The program counter (PC) is saved to the stack. At this time, the start address of the instruction to be executed after the last executed instruction is in PC and is saved when a register bank overflow occurs. The start address of the executed RESBANK instruction is in PC and is saved when a register bank underflow occurs. To prevent multiple interrupts from occurring at a register bank overflow, the interrupt priority level that caused the register bank overflow is written to the interrupt mask level bits (I3 to I0) of the status register (SR).
4. Program execution starts from the exception handler start address.

8.9 Data Transfer with Interrupt Request Signals

Interrupt request signals can be used to activate the DMAC or A-DMAC for data transfer.

Interrupt sources that are set to activate the DMAC or A-DMAC are masked without being received in the INTC. The respective mask conditions for the DMAC and A-DMAC are shown below.

(1) Mask Condition for DMAC

Mask condition = DME • (DE0 • interrupt source select 0 + DE1 • interrupt source select 1
+ DE2 • interrupt source select 2 + DE3 • interrupt source select 3
+ DE4 • interrupt source select 4 + DE5 • interrupt source select 5
+ DE6 • interrupt source select 6 + DE7 • interrupt source select 7)

Where DME is bit 0 in DMAOR of the DMAC and DEn (n = 0 to 7) is bit n in CHCR0 to CHCR7 of the DMAC.

(2) Mask Condition for A-DMAC

Mask condition = NMI + (DME • DEn) ; For ADC or ATU-III

Mask condition = NMI + (DME • (TCRn + TEn)) ; For RSPI or SCI

Where NMI is bit 1 in DMAOR of the DMAC, DME is bit 0 in ADMAOR of the A-DMAC, DEn is a bit in ADMADE corresponding to channel n of the A-DMAC, TCRn is a bit in ADMATCR corresponding to channel n of the A-DMAC, and TEn is a bit in ADMATE corresponding to channel n of the A-DMAC. TCRn should not be 0.

Note that RCAN-TL1 transfer function by the A-DMAC does not use an interrupt request signal from the RCAN-TL1. Accordingly, even if it is specified to activate the A-DMAC, it is not masked to be received in the INTC.

The DMAC or the A-DMAC clears the interrupt source flag of the interrupt request source after data transfer in response to the interrupt request signal.

For details, see section 11, Direct Memory Access Controller (DMAC), and section 12, Automotive Direct Memory Access Controller (A-DMAC).

Figure 8.13 is a block diagram of interrupt control.

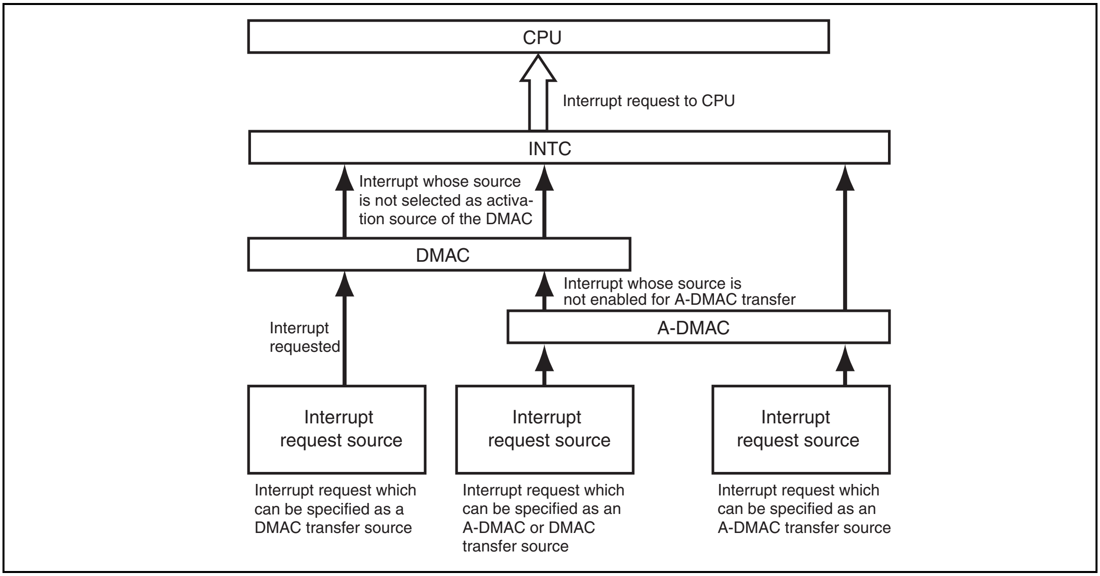

CPU
Interrupt request to CPU
INTC
Interrupt whose source is not selected as activa-tion source of the DMAC
DMAC
Interrupt whose source is not enabled for A-DMAC transfer
A-DMAC
Interrupt requested
Interrupt request source
Interrupt request source
Interrupt request source
Interrupt request which can be specified as a DMAC transfer source
Interrupt request which can be specified as an A-DMAC or DMAC transfer source
Interrupt request which can be specified as an A-DMAC transfer source

Figure 8.13 Interrupt Control Block Diagram

8.9.1 Interrupt Request Signals as Sources for CPU (Not for Activating DMAC)

1. Do not select DMAC activating sources or clear the DME bit to 0. If DMAC activating sources are selected, clear the DE bit to 0 for the relevant channel of the DMAC.
2. When an interrupt occurs, the interrupt is requested to the CPU.
3. Clear the interrupt source in the interrupt exception handler. The CPU executes the necessary processing.

8.9.2 Interrupt Request Signals as Sources for Activating DMAC (Not for CPU)

1. Selects the activation source in the DMAC and set both the DME and DE bits to 1. This settings mask CPU interrupt sources masked regardless of the interrupt priority register settings.
2. An activating signal is sent to the DMAC when an interrupt occurs.
3. The DMAC clears the interrupt source when starting transfer.

8.9.3 Interrupt Request Signals as Sources for Activating A-DMAC (Not for CPU)

1. Set the DME bit in the A-DMAC to 1, and the DE bit in the appropriate channel to 1 or ADMATCR to other than 0 (the number of transfers). This settings mask CPU interrupt sources masked regardless of the interrupt priority register settings.
2. An activating signal is sent to the A-DMAC when an interrupt occurs.
3. The A-DMAC clears the interrupt sources when starting transfer.

8.10 Usage Note

8.10.1 Timing to Clear Interrupt Source

Clear the interrupt source flag to 0 in the interrupt exception handler. The time described as "Time from occurrence of interrupt request until interrupt controller identifies priority, compares it with mask bits in SR, and sends interrupt request signal to CPU" in table 8.5 is required until the interrupt is cleared in the CPU after clearing the interrupt source flag to 0. Perform dummy read the interrupt source flag after clearing it to ensure that the interrupt request that should have been cleared is not received again erroneously. After that, the RTE instruction can be executed and the interrupt is not received again erroneously. To change interrupt levels using the LDC instruction, execute NOP at least three times after the dummy read of the interrupt source flag, and then execute the LDC instruction.

Section 9 User Break Controller (UBC)

The user break controller (UBC) provides functions that simplify program debugging. These functions make it easy to design an effective self-monitoring debugger, enabling the chip to debug programs without using an in-circuit emulator. Instruction fetch or data read/write (bus master (CPU, DMAC, or A-DMAC) selection in the case of data read/write), data size, address value, and stop timing in the case of instruction fetch are break conditions that can be set in the UBC. Since this LSI uses a Harvard architecture, instruction fetch on the CPU bus (C bus) is performed by issuing bus cycles on the instruction fetch bus (F bus), and data access on the C bus is performed by issuing bus cycles on the memory access bus (M bus). The UBC monitors the C bus and internal bus (I bus).

9.1 Features

1. The following break comparison conditions can be set.

   Number of break channels: eight channels (channels 0 to 7)

   User break can be requested as the independent condition on channels 0 to 7, respectively.

- Address

  Comparison of the 32-bit address is maskable in 1-bit units.

  One of the three address buses (F address bus (FAB), M address bus (MAB), and I address bus (IAB)) can be selected.

- Bus master when I bus is selected

  Selection of CPU cycles, DMAC cycles, or A-DMAC cycles

- Bus cycle

  Instruction fetch (only when C bus is selected) or data access

- Read/write

- Operand size

  Byte, word, and longword

2. A user-designed user-break condition exception handling routine can be run.
3. In an instruction fetch cycle, it can be selected whether a break is set before or after an instruction is executed.
4. When a break condition is satisfied, a trigger signal is output from the $\overline{\text{UBCTRG}}$ pin.

Figure 9.1 and 9.2 show a block diagram of the UBC.

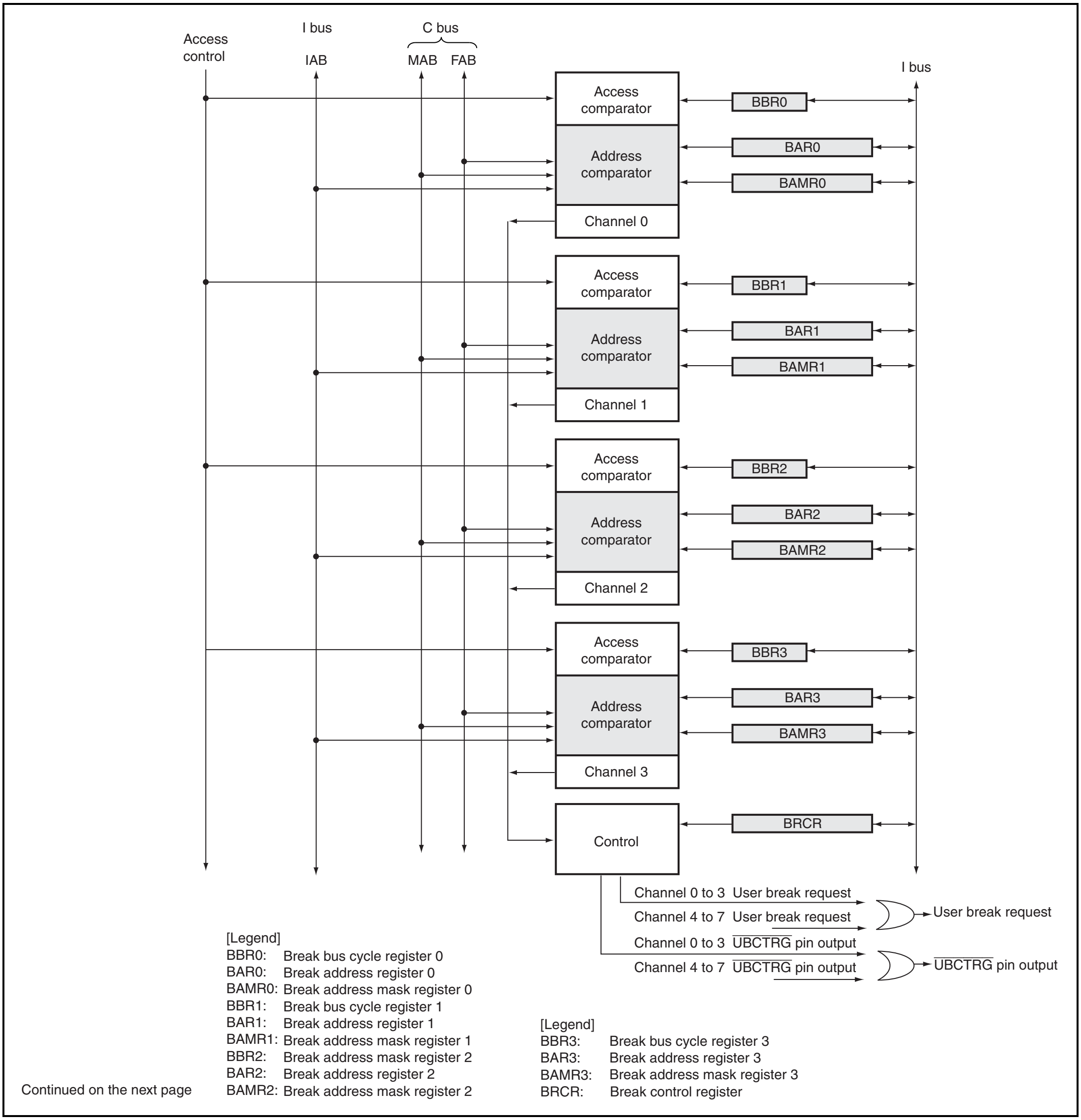

Access control
I bus
IAB
C bus
MAB
FAB
I bus
Access comparator
Address comparator
Channel 0
BBR0
BAR0
BAMR0
Access comparator
Address comparator
Channel 1
BBR1
BAR1
BAMR1
Access comparator
Address comparator
Channel 2
BBR2
BAR2
BAMR2
Access comparator
Address comparator
Channel 3
BBR3
BAR3
BAMR3
Control
BRCR
Channel 0 to 3 User break request
Channel 4 to 7 User break request
User break request
Channel 0 to 3 UBCTRG pin output
Channel 4 to 7 UBCTRG pin output
UBCTRG pin output
[Legend]
BBR0: Break bus cycle register 0
BAR0: Break address register 0
BAMR0: Break address mask register 0
BBR1: Break bus cycle register 1
BAR1: Break address register 1
BAMR1: Break address mask register 1
BBR2: Break address mask register 2
BAR2: Break address register 2
BAMR2: Break address mask register 2
[Legend]
BBR3: Break bus cycle register 3
BAR3: Break address register 3
BAMR3: Break address mask register 3
BRCR: Break control register
Continued on the next page

Figure 9.1 Block Diagram of UBC

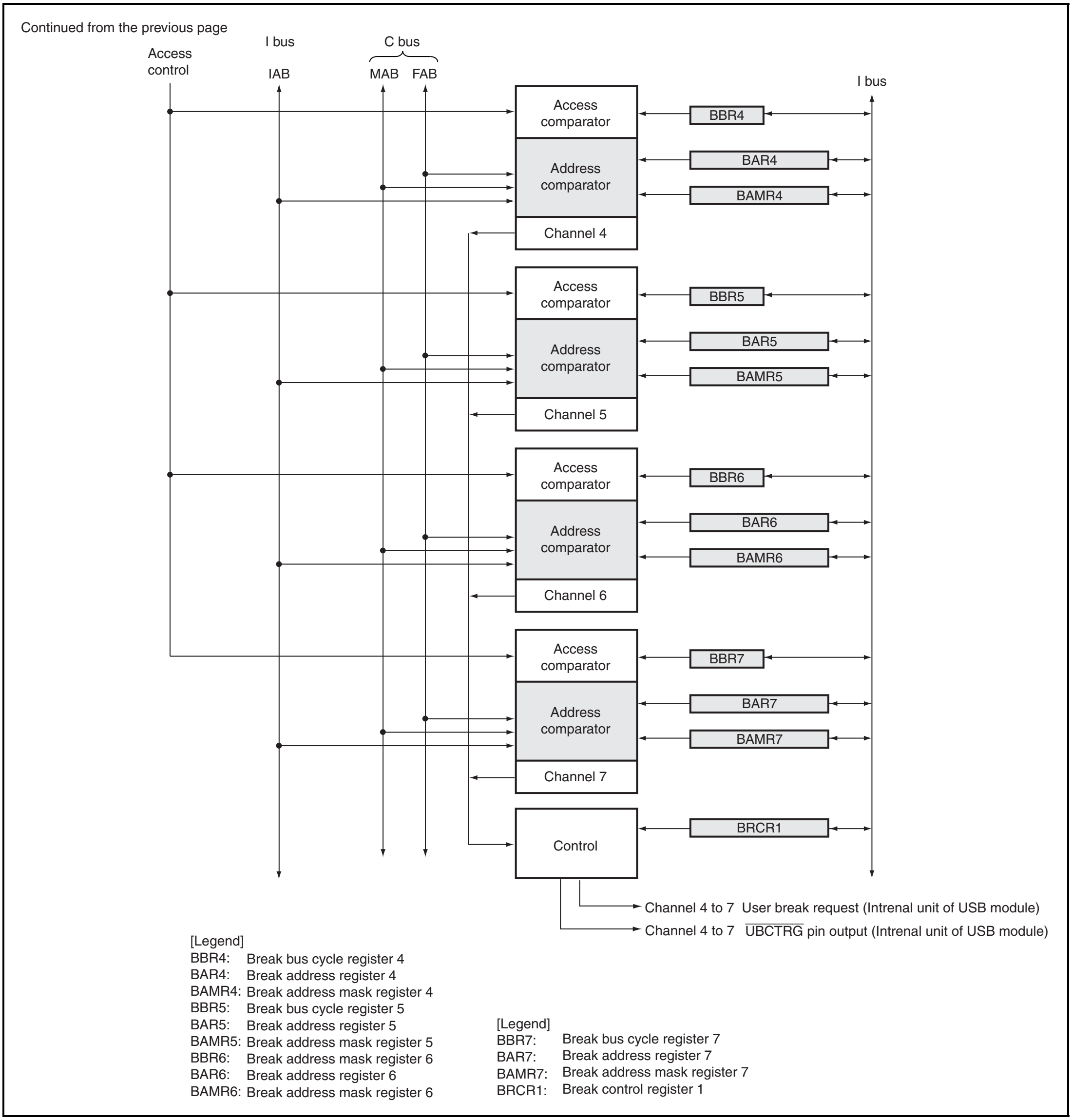

Continued from the previous page
Access control
I bus
IAB
C bus
MAB
FAB
I bus
Access comparator
Address comparator
Channel 4
BBR4
BAR4
BAMR4
Access comparator
Address comparator
Channel 5
BBR5
BAR5
BAMR5
Access comparator
Address comparator
Channel 6
BBR6
BAR6
BAMR6
Access comparator
Address comparator
Channel 7
BBR7
BAR7
BAMR7
Control
BRCR1
Channel 4 to 7 User break request (Intrenal unit of USB module)
Channel 4 to 7 $\overline{\text{UBCTRG}}$ pin output (Intrenal unit of USB module)
[Legend]
BBR4: Break bus cycle register 4
BAR4: Break address register 4
BAMR4: Break address mask register 4
BBR5: Break bus cycle register 5
BAR5: Break address register 5
BAMR5: Break address mask register 5
BBR6: Break address mask register 6
BAR6: Break address register 6
BAMR6: Break address mask register 6
[Legend]
BBR7: Break bus cycle register 7
BAR7: Break address register 7
BAMR7: Break address mask register 7
BRCR1: Break control register 1

Figure 9.2 Block Diagram of UBC 2

9.2 Input/Output Pin

Table 9.1 shows the pin configuration of the UBC.

Table 9.1 Pin Configuration

Pin Name	Symbol	I/O	Function
UBC trigger	$\overline{\text{UBCTRG}}$	Output	Indicates that a setting condition is satisfied on either of channels 0 to 7 of the UBC.

9.3 Register Descriptions

The UBC has the following registers.

Table 9.2 Register Configuration

Channel	Register Name	Symbol	R/W	Initial Value	Address	Access Size
0	Break address register 0	BAR0	R/W	H'00000000	H'FFFC0400	32
	Break address mask register 0	BAMR0	R/W	H'00000000	H'FFFC0404	32
	Break bus cycle register 0	BBR0	R/W	H'0000	H'FFFC04A0	16
1	Break address register 1	BAR1	R/W	H'00000000	H'FFFC0410	32
	Break address mask register 1	BAMR1	R/W	H'00000000	H'FFFC0414	32
	Break bus cycle register 1	BBR1	R/W	H'0000	H'FFFC04B0	16
2	Break address register 2	BAR2	R/W	H'00000000	H'FFFC0420	32
	Break address mask register 2	BAMR2	R/W	H'00000000	H'FFFC0424	32
	Break bus cycle register 2	BBR2	R/W	H'0000	H'FFFC04A4	16
3	Break address register 3	BAR3	R/W	H'00000000	H'FFFC0430	32
	Break address mask register 3	BAMR3	R/W	H'00000000	H'FFFC0434	32
	Break bus cycle register 3	BBR3	R/W	H'0000	H'FFFC04B4	16
Common (Channel 0 to 3)	Break control register	BRCR	R/W	H'00000000	H'FFFC04C0	8, 32
4	Break address register 4	BAR4	R/W	H'00000000	H'FFFC0500	32
	Break address mask register 4	BAMR4	R/W	H'00000000	H'FFFC0504	32
	Break bus cycle register 4	BBR4	R/W	H'0000	H'FFFC05A0	16
5	Break address register 5	BAR5	R/W	H'00000000	H'FFFC0510	32
	Break address mask register 5	BAMR5	R/W	H'00000000	H'FFFC0514	32
	Break bus cycle register 5	BBR5	R/W	H'0000	H'FFFC05B0	16
6	Break address register 6	BAR6	R/W	H'00000000	H'FFFC0520	32
	Break address mask register 6	BAMR6	R/W	H'00000000	H'FFFC0524	32
	Break bus cycle register 6	BBR6	R/W	H'0000	H'FFFC05A4	16

Channel	Register Name	Symbol	R/W	Initial Value	Address	Access Size
7	Break address register 7	BAR7	R/W	H'00000000	H'FFFC0530	32
	Break address mask register 7	BAMR7	R/W	H'00000000	H'FFFC0534	32
	Break bus cycle register 7	BBR7	R/W	H'0000	H'FFFC05B4	16
Common (Channel 4 to 7)	Break control register 1	BRCR1	R/W	H'00000000	H'FFFC05C0	8, 32

9.3.1 Break Address Register 0 (BAR0)

BAR0 is a 32-bit readable/writable register. BAR0 specifies the address used as a break condition in channel 0. BAR0 is initialized to H'00000000 by a power-on reset, but retains its previous value in sleep mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BA0_31	BA0_30	BA0_29	BA0_28	BA0_27	BA0_26	BA0_25	BA0_24	BA0_23	BA0_22	BA0_21	BA0_20	BA0_19	BA0_18	BA0_17	BA0_16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BA0_15	BA0_14	BA0_13	BA0_12	BA0_11	BA0_10	BA0_9	BA0_8	BA0_7	BA0_6	BA0_5	BA0_4	BA0_3	BA0_2	BA0_1	BA0_0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BA0_31 to BA0_0	All 0	R/W	Break Address 0 Store an address on the CPU address bus (FAB or MAB) or IAB specifying break conditions of channel 0. When the C bus and instruction fetch cycle are selected by BBR 0, specify an FAB address in bits BA0_31 to BA0_0. When the C bus and data access cycle are selected by BBR 0, specify an MAB address in bits BA0_31 to BA0_0.

Note: When setting the instruction fetch cycle as a break condition, clear the LSB in BAR_0 to 0.

9.3.2 Break Address Mask Register 0 (BAMR0)

BAMR0 is a 32-bit readable/writable register. BAMR0 specifies bits masked in the break address bits specified by BAR0. BAMR0 is initialized to H'00000000 by a power-on reset, but retains its previous value in sleep mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BAM0_31	BAM0_30	BAM0_29	BAM0_28	BAM0_27	BAM0_26	BAM0_25	BAM0_24	BAM0_23	BAM0_22	BAM0_21	BAM0_20	BAM0_19	BAM0_18	BAM0_17	BAM0_16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BAM0_15	BAM0_14	BAM0_13	BAM0_12	BAM0_11	BAM0_10	BAM0_9	BAM0_8	BAM0_7	BAM0_6	BAM0_5	BAM0_4	BAM0_3	BAM0_2	BAM0_1	BAM0_0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BAM0_31 to BAM0_0	All 0	R/W	Break Address Mask 0 Specify bits masked in the channel 0 break address bits specified by BAR0 (BA0_31 to BA0_0). 0: Break address bit BA0 n is included in the break condition 1: Break address bit BA0 n is masked and not included in the break condition Note: n = 31 to 0

9.3.3 Break Bus Cycle Register 0 (BBR0)

BBR0 is a 16-bit readable/writable register, which specifies (1) disabling or enabling of user break interrupts, (2) bus master of the I bus, (3) C bus cycle or I bus cycle, (4) instruction fetch or data access, (5) read or write, and (6) operand size as the break conditions of channel 0. BBR0 is initialized to H'0000 by a power-on reset, but retains its previous value in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	UBID0	-	-	CP0_ [2:0]			CD0_ [1:0]		ID0_ [1:0]		RW0_ [1:0]		SZ0_ [1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	UBID0	0	R/W	User Break Interrupt Disable 0 Disables or enables user break interrupt requests when a channel 0 break condition is satisfied. 0: User break interrupt requests enabled 1: User break interrupt requests disabled
12, 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 8	CP0_[2:0]	000	R/W	I-Bus Bus Master Select 0 Select the bus master when the bus cycle of the channel 0 break condition is the I bus cycle. However, when the C bus cycle is selected, this bit is invalidated (only the CPU cycle). xx1: CPU cycle is included in break conditions x1x: DMAC cycle is included in break conditions 1xx: A-DMAC cycle is included in break conditions

Bit	Bit Name	Initial Value	R/W	Description
7, 6	CD0_[1:0]	00	R/W	C Bus Cycle/I Bus Cycle Select 0 Select the C bus cycle or I bus cycle as the bus cycle of the channel 0 break condition. 00: Condition comparison is not performed 01: Break condition is the C bus (F bus or M bus) cycle 10: Break condition is the I bus cycle 11: Break condition is the C bus (F bus or M bus) cycle
5, 4	ID0_[1:0]	00	R/W	Instruction Fetch/Data Access Select 0 Select the instruction fetch cycle or data access cycle as the bus cycle of the channel 0 break condition. If the instruction fetch cycle is selected, select the C bus cycle. 00: Condition comparison is not performed 01: Break condition is the instruction fetch cycle 10: Break condition is the data access cycle 11: Break condition is the instruction fetch cycle or data access cycle
3, 2	RW0_[1:0]	00	R/W	Read/Write Select 0 Select the read cycle or write cycle as the bus cycle of the channel 0 break condition. 00: Condition comparison is not performed 01: Break condition is the read cycle 10: Break condition is the write cycle 11: Break condition is the read cycle or write cycle
1, 0	SZ0_[1:0]	00	R/W	Operand Size Select 0 Select the operand size of the bus cycle for the channel 0 break condition. 00: Break condition does not include operand size 01: Break condition is byte access 10: Break condition is word access 11: Break condition is longword access

[Legend]
x: Don't care

9.3.4 Break Address Register 1 (BAR1)

BAR1 is a 32-bit readable/writable register. BAR1 specifies the address used as a break condition in channel 1. The control bits CD1_1 and CD1_0 in the break bus cycle register 1 (BBR1) select one of the two address buses for a channel 1 break condition. BAR1 is initialized to H'00000000 by a power-on reset, but retains its previous value in sleep mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BA1_31	BA1_30	BA1_29	BA1_28	BA1_27	BA1_26	BA1_25	BA1_24	BA1_23	BA1_22	BA1_21	BA1_20	BA1_19	BA1_18	BA1_17	BA1_16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BA1_15	BA1_14	BA1_13	BA1_12	BA1_11	BA1_10	BA1_9	BA1_8	BA1_7	BA1_6	BA1_5	BA1_4	BA1_3	BA1_2	BA1_1	BA1_0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BA1_31 to BA1_0	All 0	R/W	Break Address 1 Store the CPU address bus (FAB or MAB) or IAB address specifying break conditions of channel 1. When the C bus and instruction fetch cycle are selected by BBR1, specify an FAB address in bits BA1_31 to BA1_0. When the C bus and data access cycle are selected by BBR1, specify an MAB address in bits BA1_31 to BA1_0.

Note: When setting the instruction fetch cycle as a break condition, clear the LSB in BAR1 to 0.

9.3.5 Break Address Mask Register 1 (BAMR1)

BAMR1 is a 32-bit readable/writable register. BAMR1 specifies bits masked in the break address bits specified by BAR1. BAMR1 is initialized to H'00000000 by a power-on reset, but retains its previous value in sleep mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BAM1_31	BAM1_30	BAM1_29	BAM1_28	BAM1_27	BAM1_26	BAM1_25	BAM1_24	BAM1_23	BAM1_22	BAM1_21	BAM1_20	BAM1_19	BAM1_18	BAM1_17	BAM1_16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BAM1_15	BAM1_14	BAM1_13	BAM1_12	BAM1_11	BAM1_10	BAM1_9	BAM1_8	BAM1_7	BAM1_6	BAM1_5	BAM1_4	BAM1_3	BAM1_2	BAM1_1	BAM1_0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BAM1_31 to BAM1_0	All 0	R/W	Break Address Mask 1 Specify bits masked in the channel 1 break address bits specified by BAR1 (BA1_31 to BA1_0). 0: Break address bit BA1_n is included in the break condition 1: Break address bit BA1_n is masked and not included in the break condition Note: n = 31 to 0

9.3.6 Break Bus Cycle Register 1 (BBR1)

BBR1 is a 16-bit readable/writable register, which specifies (1) disabling or enabling of user break interrupts, (2) bus master of the I bus, (3) C bus cycle or I bus cycle, (4) instruction fetch or data access, (5) read or write, and (6) operand size as the break conditions of channel 1. BBR1 is initialized to H'0000 by a power-on reset, but retains its previous value in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	UBID1	-	-	CP1_ [2:0]			CD1_ [1:0]		ID1_ [1:0]		RW1_ [1:0]		SZ1_ [1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	UBID1	0	R/W	User Break Interrupt Disable 1 Disables or enables user break interrupt requests when a channel 1 break condition is satisfied. 0: User break interrupt requests enabled 1: User break interrupt requests disabled
12, 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 8	CP1_[2:0]	000	R/W	I-Bus Bus Master Select 1 Select the bus master when the bus cycle of the channel 1 break condition is the I bus cycle. However, when the C bus cycle is selected, this bit is invalidated (only the CPU cycle). xx1: CPU cycle is included in break conditions x1x: DMAC cycle is included in break conditions 1xx: A-DMAC cycle is included in break conditions

Bit	Bit Name	Initial Value	R/W	Description
7, 6	CD1_[1:0]	00	R/W	C Bus Cycle/I Bus Cycle Select 1 Select the C bus cycle or I bus cycle as the bus cycle of the channel 1 break condition. 00: Condition comparison is not performed 01: Break condition is the C bus cycle 10: Break condition is the I bus cycle 11: Break condition is the C bus cycle
5, 4	ID1_[1:0]	00	R/W	Instruction Fetch/Data Access Select 1 Select the instruction fetch cycle or data access cycle as the bus cycle of the channel 1 break condition. If the instruction fetch cycle is selected, select the C bus cycle. 00: Condition comparison is not performed 01: Break condition is the instruction fetch cycle 10: Break condition is the data access cycle 11: Break condition is the instruction fetch cycle or data access cycle
3, 2	RW1_[1:0]	00	R/W	Read/Write Select 1 Select the read cycle or write cycle as the bus cycle of the channel 1 break condition. 00: Condition comparison is not performed 01: Break condition is the read cycle 10: Break condition is the write cycle 11: Break condition is the read cycle or write cycle
1, 0	SZ1_[1:0]	00	R/W	Operand Size Select 1 Select the operand size of the bus cycle for the channel 1 break condition. 00: Break condition does not include operand size 01: Break condition is byte access 10: Break condition is word access 11: Break condition is longword access

[Legend]

x: Don't care

9.3.7 Break Address Register 2 (BAR2)

BAR2 is a 32-bit readable/writable register. BAR2 specifies the address used as a break condition in channel 2. The control bits CD2_1 and CD2_0 in the break bus cycle register 2 (BBR2) select one of the two address buses for a channel 2 break condition. BAR2 is initialized to H'00000000 by a power-on reset, but retains its previous value in sleep mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BA2_31	BA2_30	BA2_29	BA2_28	BA2_27	BA2_26	BA2_25	BA2_24	BA2_23	BA2_22	BA2_21	BA2_20	BA2_19	BA2_18	BA2_17	BA2_16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BA2_15	BA2_14	BA2_13	BA2_12	BA2_11	BA2_10	BA2_9	BA2_8	BA2_7	BA2_6	BA2_5	BA2_4	BA2_3	BA2_2	BA2_1	BA2_0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BA2_31 to BA2_0	All 0	R/W	Break Address 2 Store an address on the CPU address bus (FAB or MAB) or IAB specifying break conditions of channel 2. When the C bus and instruction fetch cycle are selected by BBR2, specify an FAB address in bits BA2_31 to BA2_0. When the C bus and data access cycle are selected by BBR2, specify an MAB address in bits BA2_31 to BA2_0.

Note: When setting the instruction fetch cycle as a break condition, clear the LSB in BAR2 to 0.

9.3.8 Break Address Mask Register 2 (BAMR2)

BAMR2 is a 32-bit readable/writable register. BAMR2 specifies bits masked in the break address bits specified by BAR2. BAMR2 is initialized to H'00000000 by a power-on reset, but retains its previous value in sleep mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BAM2_31	BAM2_30	BAM2_29	BAM2_28	BAM2_27	BAM2_26	BAM2_25	BAM2_24	BAM2_23	BAM2_22	BAM2_21	BAM2_20	BAM2_19	BAM2_18	BAM2_17	BAM2_16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BAM2_15	BAM2_14	BAM2_13	BAM2_12	BAM2_11	BAM2_10	BAM2_9	BAM2_8	BAM2_7	BAM2_6	BAM2_5	BAM2_4	BAM2_3	BAM2_2	BAM2_1	BAM2_0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BAM2_31 to BAM2_0	All 0	R/W	Break Address Mask 2 Specify bits masked in the channel 2 break address bits specified by BAR2 (BA2_31 to BA2_0). 0: Break address bit BA2_n is included in the break condition 1: Break address bit BA2_n is masked and not included in the break condition Note: n = 31 to 0

9.3.9 Break Bus Cycle Register 2 (BBR2)

BBR2 is a 16-bit readable/writable register, which specifies (1) disabling or enabling of user break interrupts, (2) bus master of the I bus, (3) C bus cycle or I bus cycle, (4) instruction fetch or data access, (5) read or write, and (6) operand size as the break conditions of channel 2. BBR2 is initialized to H'0000 by a power-on reset, but retains its previous value in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	UBID2	-	-	CP2_ [2:0]			CD2_ [1:0]		ID2_ [1:0]		RW2_ [1:0]		SZ2_ [1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	UBID2	0	R/W	User Break Interrupt Disable 2 Disables or enables user break interrupt requests when a channel 2 break condition is satisfied. 0: User break interrupt requests enabled 1: User break interrupt requests disabled
12, 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 8	CP2_[2:0]	000	R/W	I-Bus Bus Master Select 2 Select the bus master when the bus cycle of the channel 2 break condition is the I bus cycle. However, when the C bus cycle is selected, this bit is invalidated (only the CPU cycle). xx1: CPU cycle is included in break conditions x1x: DMAC cycle is included in break conditions 1xx: A-DMAC cycle is included in break conditions

Bit	Bit Name	Initial Value	R/W	Description
7, 6	CD2_[1:0]	00	R/W	C Bus Cycle/I Bus Cycle Select 2 Select the C bus cycle or I bus cycle as the bus cycle of the channel 2 break condition. 00: Condition comparison is not performed 01: Break condition is the C bus cycle 10: Break condition is the I bus cycle 11: Break condition is the C bus cycle
5, 4	ID2_[1:0]	00	R/W	Instruction Fetch/Data Access Select 2 Select the instruction fetch cycle or data access cycle as the bus cycle of the channel 2 break condition. If the instruction fetch cycle is selected, select the C bus cycle. 00: Condition comparison is not performed 01: Break condition is the instruction fetch cycle 10: Break condition is the data access cycle 11: Break condition is the instruction fetch cycle or data access cycle
3, 2	RW2_[1:0]	00	R/W	Read/Write Select 2 Select the read cycle or write cycle as the bus cycle of the channel 2 break condition. 00: Condition comparison is not performed 01: Break condition is the read cycle 10: Break condition is the write cycle 11: Break condition is the read cycle or write cycle
1, 0	SZ2_[1:0]	00	R/W	Operand Size Select 2 Select the operand size of the bus cycle for the channel 2 break condition. 00: Break condition does not include operand size 01: Break condition is byte access 10: Break condition is word access 11: Break condition is longword access

[Legend]

x: Don't care

9.3.10 Break Address Register 3 (BAR3)

BAR3 is a 32-bit readable/writable register. BAR3 specifies the address used as a break condition in channel 3. The control bits CD3_1 and CD3_0 in the break bus cycle register 3 (BBR3) select one of the two address buses for a channel 3 break condition. BAR3 is initialized to H'00000000 by a power-on reset, but retains its previous value in sleep mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BA3_31	BA3_30	BA3_29	BA3_28	BA3_27	BA3_26	BA3_25	BA3_24	BA3_23	BA3_22	BA3_21	BA3_20	BA3_19	BA3_18	BA3_17	BA3_16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BA3_15	BA3_14	BA3_13	BA3_12	BA3_11	BA3_10	BA3_9	BA3_8	BA3_7	BA3_6	BA3_5	BA3_4	BA3_3	BA3_2	BA3_1	BA3_0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BA3_31 to BA3_0	All 0	R/W	Break Address 3 Store the CPU address bus (FAB or MAB) or IAB address specifying break conditions of channel 3. When the C bus and instruction fetch cycle are selected by BBR3, specify an FAB address in bits BA3_31 to BA3_0. When the C bus and data access cycle are selected by BBR3, specify an MAB address in bits BA3_31 to BA3_0.

Note: When setting the instruction fetch cycle as a break condition, clear the LSB in BAR3 to 0.

9.3.11 Break Address Mask Register 3 (BAMR3)

BAMR3 is a 32-bit readable/writable register. BAMR3 specifies bits masked in the break address bits specified by BAR3. BAMR3 is initialized to H'00000000 by a power-on reset, but retains its previous value in sleep mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BAM3_31	BAM3_30	BAM3_29	BAM3_28	BAM3_27	BAM3_26	BAM3_25	BAM3_24	BAM3_23	BAM3_22	BAM3_21	BAM3_20	BAM3_19	BAM3_18	BAM3_17	BAM3_16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BAM3_15	BAM3_14	BAM3_13	BAM3_12	BAM3_11	BAM3_10	BAM3_9	BAM3_8	BAM3_7	BAM3_6	BAM3_5	BAM3_4	BAM3_3	BAM3_2	BAM3_1	BAM3_0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BAM3_31 to BAM3_0	All 0	R/W	Break Address Mask 3 Specify bits masked in the channel 3 break address bits specified by BAR3 (BA3_31 to BA3_0). 0: Break address bit BA3_n is included in the break condition 1: Break address bit BA3_n is masked and not included in the break condition Note: n = 31 to 0

9.3.12 Break Bus Cycle Register 3 (BBR3)

BBR3 is a 16-bit readable/writable register, which specifies (1) disabling or enabling of user break interrupts, (2) bus master of the I bus, (3) C bus cycle or I bus cycle, (4) instruction fetch or data access, (5) read or write, and (6) operand size as the break conditions of channel 3. BBR3 is initialized to H'0000 by a power-on reset, but retains its previous value in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	UBID3	-	-	CP3_ [2:0]			CD3_ [1:0]		ID3_ [1:0]		RW3_ [1:0]		SZ3_ [1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	UBID3	0	R/W	User Break Interrupt Disable 3 Disables or enables user break interrupt requests when a channel 3 break condition is satisfied. 0: User break interrupt requests enabled 1: User break interrupt requests disabled
12, 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 8	CP3_[2:0]	000	R/W	I-Bus Bus Master Select 3 Select the bus master when the bus cycle of the channel 3 break condition is the I bus cycle. However, when the C bus cycle is selected, this bit is invalidated (only the CPU cycle). xx1: CPU cycle is included in break conditions x1x: DMAC cycle is included in break conditions 1xx: A-DMAC cycle is included in break conditions

Bit	Bit Name	Initial Value	R/W	Description
7, 6	CD3_[1:0]	00	R/W	C Bus Cycle/I Bus Cycle Select 3 Select the C bus cycle or I bus cycle as the bus cycle of the channel 3 break condition. 00: Condition comparison is not performed 01: Break condition is the C bus cycle 10: Break condition is the I bus cycle 11: Break condition is the C bus cycle
5, 4	ID3_[1:0]	00	R/W	Instruction Fetch/Data Access Select 3 Select the instruction fetch cycle or data access cycle as the bus cycle of the channel 3 break condition. If the instruction fetch cycle is selected, select the C bus cycle. 00: Condition comparison is not performed 01: Break condition is the instruction fetch cycle 10: Break condition is the data access cycle 11: Break condition is the instruction fetch cycle or data access cycle
3, 2	RW3_[1:0]	00	R/W	Read/Write Select 3 Select the read cycle or write cycle as the bus cycle of the channel 3 break condition. 00: Condition comparison is not performed 01: Break condition is the read cycle 10: Break condition is the write cycle 11: Break condition is the read cycle or write cycle
1, 0	SZ3_[1:0]	00	R/W	Operand Size Select 3 Select the operand size of the bus cycle for the channel 3 break condition. 00: Break condition does not include operand size 01: Break condition is byte access 10: Break condition is word access 11: Break condition is longword access

[Legend]

x: Don't care

9.3.13 Break Control Register (BRCR)

BRCR sets the following conditions:

1. Specifies whether PC breaks on channels 0 to 3 are set before or after instruction execution.
2. Specifies the pulse width of the $\overline{\text{UBCTRG}}$ output when break conditions on channels 0 to 7 are satisfied.
3. Specifies whether or not the $\overline{\text{UBCTRG}}$ output is disabled when break conditions on channels 0 to 3 are satisfied.

BRCR is a 32-bit readable/writable register that has break condition match flags and bits for setting other break conditions. For the condition match flags of bits 15 to 8, writing 1 is invalid (previous values are retained) and writing 0 is only possible. To clear the flag, write 0 to the flag bit to be cleared and 1 to all other flag bits. BRCR is initialized to H'00000000 by a power-on reset, but retains its previous value in sleep mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	UTOD3	UTOD2	UTOD1	UTOD0	CKS[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SCMFC0	SCMFC1	SCMFC2	SCMFC3	SCMFD0	SCMFD1	SCMFD2	SCMFD3	PCB3	PCB2	PCB1	PCB0	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
21	UTOD3	0	R/W	$\overline{\text{UBCTRG}}$ Output Disable 3 Specifies whether or not $\overline{\text{UBCTRG}}$ output is disabled when a break condition is satisfied. 0: $\overline{\text{UBCTRG}}$ output is not disabled when a break condition for channel 3 is satisfied. 1: $\overline{\text{UBCTRG}}$ output is disabled when a break condition for channel 3 is satisfied.

Bit	Bit Name	Initial Value	R/W	Description
20	UTOD2	0	R/W	$\overline{\text{UBCTRG}}$ Output Disable 2 Specifies whether or not $\overline{\text{UBCTRG}}$ output is disabled when a break condition is satisfied. 0: $\overline{\text{UBCTRG}}$ output is not disabled when a break condition for channel 2 is satisfied. 1: $\overline{\text{UBCTRG}}$ output is disabled when a break condition for channel 2 is satisfied.
19	UTOD1	0	R/W	$\overline{\text{UBCTRG}}$ Output Disable 1 Specifies whether or not $\overline{\text{UBCTRG}}$ output is disabled when a break condition is satisfied. 0: $\overline{\text{UBCTRG}}$ output is not disabled when a break condition for channel 1 is satisfied. 1: $\overline{\text{UBCTRG}}$ output is disabled when a break condition for channel 1 is satisfied.
18	UTOD0	0	R/W	$\overline{\text{UBCTRG}}$ Output Disable 0 Specifies whether or not $\overline{\text{UBCTRG}}$ output is disabled when a break condition is satisfied. 0: $\overline{\text{UBCTRG}}$ output is not disabled when a break condition for channel 0 is satisfied. 1: $\overline{\text{UBCTRG}}$ output is disabled when a break condition for channel 0 is satisfied.
17, 16	CKS[1:0]	00	R/W	Clock Select Specifies the pulse width output to the $\overline{\text{UBCTRG}}$ pin when a break condition is satisfied. 00: Pulse width of $\overline{\text{UBCTRG}}$ is one Pϕ cycle (prohibited when two-time multiplication peripheral clock is set) 01: Pulse width of $\overline{\text{UBCTRG}}$ is two Pϕ cycles 10: Pulse width of $\overline{\text{UBCTRG}}$ is four Pϕ cycles 11: Pulse width of $\overline{\text{UBCTRG}}$ is eight Pϕ cycles
15	SCMFC0	0	R/W	C Bus Cycle Condition Match Flag 0 When the C bus cycle condition in the break conditions set for channel 0 is satisfied, this flag is set to 1. In order to clear this flag, write 0 to this bit. 0: The C bus cycle condition for channel 0 does not match 1: The C bus cycle condition for channel 0 matches

Bit	Bit Name	Initial Value	R/W	Description
14	SCMFC1	0	R/W	C Bus Cycle Condition Match Flag 1 When the C bus cycle condition in the break conditions set for channel 1 is satisfied, this flag is set to 1. In order to clear this flag, write 0 to this bit. 0: The C bus cycle condition for channel 1 does not match 1: The C bus cycle condition for channel 1 matches
13	SCMFC2	0	R/W	C Bus Cycle Condition Match Flag 2 When the C bus cycle condition in the break conditions set for channel 2 is satisfied, this flag is set to 1. In order to clear this flag, write 0 to this bit. 0: The C bus cycle condition for channel 2 does not match 1: The C bus cycle condition for channel 2 matches
12	SCMFC3	0	R/W	C Bus Cycle Condition Match Flag 3 When the C bus cycle condition in the break conditions set for channel 3 is satisfied, this flag is set to 1. In order to clear this flag, write 0 to this bit. 0: The C bus cycle condition for channel 3 does not match 1: The C bus cycle condition for channel 3 matches
11	SCMFD0	0	R/W	I Bus Cycle Condition Match Flag 0 When the I bus cycle condition in the break conditions set for channel 0 is satisfied, this flag is set to 1. In order to clear this flag, write 0 to this bit. 0: The I bus cycle condition for channel 0 does not match 1: The I bus cycle condition for channel 0 matches
10	SCMFD1	0	R/W	I Bus Cycle Condition Match Flag 1 When the I bus cycle condition in the break conditions set for channel 1 is satisfied, this flag is set to 1. In order to clear this flag, write 0 to this bit. 0: The I bus cycle condition for channel 1 does not match 1: The I bus cycle condition for channel 1 matches

Bit	Bit Name	Initial Value	R/W	Description
9	SCMFD2	0	R/W	I Bus Cycle Condition Match Flag 2 When the I bus cycle condition in the break conditions set for channel 2 is satisfied, this flag is set to 1. In order to clear this flag, write 0 to this bit. 0: The I bus cycle condition for channel 2 does not match 1: The I bus cycle condition for channel 2 matches
8	SCMFD3	0	R/W	I Bus Cycle Condition Match Flag 3 When the I bus cycle condition in the break conditions set for channel 3 is satisfied, this flag is set to 1. In order to clear this flag, write 0 to this bit. 0: The I bus cycle condition for channel 3 does not match 1: The I bus cycle condition for channel 3 matches
7	PCB3	0	R/W	PC Break Select 3 Selects the break timing of the instruction fetch cycle for channel 3 as before or after instruction execution. 0: PC break of channel 3 is generated before instruction execution 1: PC break of channel 3 is generated after instruction execution
6	PCB2	0	R/W	PC Break Select 2 Selects the break timing of the instruction fetch cycle for channel 2 as before or after instruction execution. 0: PC break of channel 2 is generated before instruction execution 1: PC break of channel 2 is generated after instruction execution
5	PCB1	0	R/W	PC Break Select 1 Selects the break timing of the instruction fetch cycle for channel 1 as before or after instruction execution. 0: PC break of channel 1 is generated before instruction execution 1: PC break of channel 1 is generated after instruction execution

Bit	Bit Name	Initial Value	R/W	Description
4	PCB0	0	R/W	PC Break Select 0 Selects the break timing of the instruction fetch cycle for channel 0 as before or after instruction execution. 0: PC break of channel 0 is generated before instruction execution 1: PC break of channel 0 is generated after instruction execution
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

9.3.14 Break Address Register 4 (BAR4)

BAR4 is a 32-bit readable/writable register. BAR4 specifies the address used as a break condition in channel 4. BAR4 is initialized to H'00000000 by a power-on reset, but retains its previous value in sleep mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BA4_31	BA4_30	BA4_29	BA4_28	BA4_27	BA4_26	BA4_25	BA4_24	BA4_23	BA4_22	BA4_21	BA4_20	BA4_19	BA4_18	BA4_17	BA4_16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BA4_15	BA4_14	BA4_13	BA4_12	BA4_11	BA4_10	BA4_9	BA4_8	BA4_7	BA4_6	BA4_5	BA4_4	BA4_3	BA4_2	BA4_1	BA4_0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BA4_31 to BA4_0	All 0	R/W	Break Address 4 Store an address on the CPU address bus (FAB or MAB) or IAB specifying break conditions of channel 4. When the C bus and instruction fetch cycle are selected by BBR4, specify an FAB address in bits BA4_31 to BA4_0. When the C bus and data access cycle are selected by BBR4, specify an MAB address in bits BA4_31 to BA4_0.

Note: When setting the instruction fetch cycle as a break condition, clear the LSB in BAR4 to 0.

9.3.15 Break Address Mask Register 4 (BAMR4)

BAMR4 is a 32-bit readable/writable register. BAMR4 specifies bits masked in the break address bits specified by BAR4. BAMR4 is initialized to H'00000000 by a power-on reset, but retains its previous value in sleep mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BAM4_31	BAM4_30	BAM4_29	BAM4_28	BAM4_27	BAM4_26	BAM4_25	BAM4_24	BAM4_23	BAM4_22	BAM4_21	BAM4_20	BAM4_19	BAM4_18	BAM4_17	BAM4_16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BAM4_15	BAM4_14	BAM4_13	BAM4_12	BAM4_11	BAM4_10	BAM4_9	BAM4_8	BAM4_7	BAM4_6	BAM4_5	BAM4_4	BAM4_3	BAM4_2	BAM4_1	BAM4_0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BAM4_31 to BAM4_0	All 0	R/W	Break Address Mask 4 Specify bits masked in the channel 0 break address bits specified by BAR4 (BA4_31 to BA4_0). 0: Break address bit BA4_n is included in the break condition 1: Break address bit BA4_n is masked and not included in the break condition Note: n = 31 to 0

9.3.16 Break Bus Cycle Register 4 (BBR4)

BBR4 is a 16-bit readable/writable register, which specifies (1) disabling or enabling of user break interrupts, (2) bus master of the I bus, (3) C bus cycle or I bus cycle, (4) instruction fetch or data access, (5) read or write, and (6) operand size as the break conditions of channel 4. BBR4 is initialized to H'0000 by a power-on reset, but retains its previous value in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	UBID4	-	-	CP4_ [2:0]			CD4_ [1:0]		ID4_ [1:0]		RW4_ [1:0]		SZ4_ [1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	UBID4	0	R/W	User Break Interrupt Disable 4 Disables or enables user break interrupt requests when a channel 4 break condition is satisfied. 0: User break interrupt requests enabled 1: User break interrupt requests disabled
12, 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 8	CP4_[2:0]	000	R/W	I-Bus Bus Master Select 4 Select the bus master when the bus cycle of the channel 4 break condition is the I bus cycle. However, when the C bus cycle is selected, this bit is invalidated (only the CPU cycle). xx1: CPU cycle is included in break conditions x1x: DMAC cycle is included in break conditions 1xx: A-DMAC cycle is included in break conditions

Bit	Bit Name	Initial Value	R/W	Description
7, 6	CD4_[1:0]	00	R/W	C Bus Cycle/I Bus Cycle Select 4 Select the C bus cycle or I bus cycle as the bus cycle of the channel 4 break condition. 00: Condition comparison is not performed 01: Break condition is the C bus (F bus or M bus) cycle 10: Break condition is the I bus cycle 11: Break condition is the C bus (F bus or M bus) cycle
5, 4	ID4_[1:0]	00	R/W	Instruction Fetch/Data Access Select 4 Select the instruction fetch cycle or data access cycle as the bus cycle of the channel 4 break condition. If the instruction fetch cycle is selected, select the C bus cycle. 00: Condition comparison is not performed 01: Break condition is the instruction fetch cycle 10: Break condition is the data access cycle 11: Break condition is the instruction fetch cycle or data access cycle
3, 2	RW4_[1:0]	00	R/W	Read/Write Select 4 Select the read cycle or write cycle as the bus cycle of the channel 4 break condition. 00: Condition comparison is not performed 01: Break condition is the read cycle 10: Break condition is the write cycle 11: Break condition is the read cycle or write cycle
1, 0	SZ4_[1:0]	00	R/W	Operand Size Select 4 Select the operand size of the bus cycle for the channel 4 break condition. 00: Break condition does not include operand size 01: Break condition is byte access 10: Break condition is word access 11: Break condition is longword access

[Legend]

x: Don't care

9.3.17 Break Address Register 5 (BAR5)

BAR5 is a 32-bit readable/writable register. BAR5 specifies the address used as a break condition in channel 5. The control bits CD5_1 and CD5_0 in the break bus cycle register 5 (BBR1) select one of the two address buses for a channel 5 break condition. BAR5 is initialized to H'00000000 by a power-on reset, but retains its previous value in sleep mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BA5_31	BA5_30	BA5_29	BA5_28	BA5_27	BA5_26	BA5_25	BA5_24	BA5_23	BA5_22	BA5_21	BA5_20	BA5_19	BA5_18	BA5_17	BA5_16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BA5_15	BA5_14	BA5_13	BA5_12	BA5_11	BA5_10	BA5_9	BA5_8	BA5_7	BA5_6	BA5_5	BA5_4	BA5_3	BA5_2	BA5_1	BA5_0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BA5_31 to BA5_0	All 0	R/W	Break Address 5 Store the CPU address bus (FAB or MAB) or IAB address specifying break conditions of channel 5. When the C bus and instruction fetch cycle are selected by BBR5, specify an FAB address in bits BA5_31 to BA5_0. When the C bus and data access cycle are selected by BBR5, specify an MAB address in bits BA5_31 to BA5_0.

Note: When setting the instruction fetch cycle as a break condition, clear the LSB in BAR5 to 0.

9.3.18 Break Address Mask Register 5 (BAMR5)

BAMR5 is a 32-bit readable/writable register. BAMR5 specifies bits masked in the break address bits specified by BAR5. BAMR5 is initialized to H'00000000 by a power-on reset, but retains its previous value in sleep mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BAM5_31	BAM5_30	BAM5_29	BAM5_28	BAM5_27	BAM5_26	BAM5_25	BAM5_24	BAM5_23	BAM5_22	BAM5_21	BAM5_20	BAM5_19	BAM5_18	BAM5_17	BAM5_16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BAM5_15	BAM5_14	BAM5_13	BAM5_12	BAM5_11	BAM5_10	BAM5_9	BAM5_8	BAM5_7	BAM5_6	BAM5_5	BAM5_4	BAM5_3	BAM5_2	BAM5_1	BAM5_0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BAM5_31 to BAM5_0	All 0	R/W	Break Address Mask 5 Specify bits masked in the channel 5 break address bits specified by BAR5 (BA5_31 to BA5_0). 0: Break address bit BA5_n is included in the break condition 1: Break address bit BA5_n is masked and not included in the break condition Note: n = 31 to 0

9.3.19 Break Bus Cycle Register 5 (BBR5)

BBR5 is a 16-bit readable/writable register, which specifies (1) disabling or enabling of user break interrupts, (2) bus master of the I bus, (3) C bus cycle or I bus cycle, (4) instruction fetch or data access, (5) read or write, and (6) operand size as the break conditions of channel 5. BBR5 is initialized to H'0000 by a power-on reset, but retains its previous value in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	UBID5	-	-	CP5_ [2:0]			CD5_ [1:0]		ID5_ [1:0]		RW5_ [1:0]		SZ5_ [1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	UBID5	0	R/W	User Break Interrupt Disable 5 Disables or enables user break interrupt requests when a channel 5 break condition is satisfied. 0: User break interrupt requests enabled 1: User break interrupt requests disabled
12, 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 8	CP5_[2:0]	000	R/W	I-Bus Bus Master Select 5 Select the bus master when the bus cycle of the channel 5 break condition is the I bus cycle. However, when the C bus cycle is selected, this bit is invalidated (only the CPU cycle). xx1: CPU cycle is included in break conditions x1x: DMAC cycle is included in break conditions 1xx: A-DMAC cycle is included in break conditions

Bit	Bit Name	Initial Value	R/W	Description
7, 6	CD5_[1:0]	00	R/W	C Bus Cycle/I Bus Cycle Select 5 Select the C bus cycle or I bus cycle as the bus cycle of the channel 5 break condition. 00: Condition comparison is not performed 01: Break condition is the C bus cycle 10: Break condition is the I bus cycle 11: Break condition is the C bus cycle
5, 4	ID5_[1:0]	00	R/W	Instruction Fetch/Data Access Select 5 Select the instruction fetch cycle or data access cycle as the bus cycle of the channel 5 break condition. If the instruction fetch cycle is selected, select the C bus cycle. 00: Condition comparison is not performed 01: Break condition is the instruction fetch cycle 10: Break condition is the data access cycle 11: Break condition is the instruction fetch cycle or data access cycle
3, 2	RW5_[1:0]	00	R/W	Read/Write Select 5 Select the read cycle or write cycle as the bus cycle of the channel 5 break condition. 00: Condition comparison is not performed 01: Break condition is the read cycle 10: Break condition is the write cycle 11: Break condition is the read cycle or write cycle
1, 0	SZ5_[1:0]	00	R/W	Operand Size Select 5 Select the operand size of the bus cycle for the channel 5 break condition. 00: Break condition does not include operand size 01: Break condition is byte access 10: Break condition is word access 11: Break condition is longword access

[Legend]

x: Don't care

9.3.20 Break Address Register 6 (BAR6)

BAR6 is a 32-bit readable/writable register. BAR6 specifies the address used as a break condition in channel 6. The control bits CD6_1 and CD6_0 in the break bus cycle register 6 (BBR6) select one of the two address buses for a channel 6 break condition. BAR6 is initialized to H'00000000 by a power-on reset, but retains its previous value in sleep mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BA6_31	BA6_30	BA6_29	BA6_28	BA6_27	BA6_26	BA6_25	BA6_24	BA6_23	BA6_22	BA6_21	BA6_20	BA6_19	BA6_18	BA6_17	BA6_16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BA6_15	BA6_14	BA6_13	BA6_12	BA6_11	BA6_10	BA6_9	BA6_8	BA6_7	BA6_6	BA6_5	BA6_4	BA6_3	BA6_2	BA6_1	BA6_0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BA6_31 to BA6_0	All 0	R/W	Break Address 6 Store an address on the CPU address bus (FAB or MAB) or IAB specifying break conditions of channel 6. When the C bus and instruction fetch cycle are selected by BBR6, specify an FAB address in bits BA6_31 to BA6_0. When the C bus and data access cycle are selected by BBR6, specify an MAB address in bits BA6_31 to BA6_0.

Note: When setting the instruction fetch cycle as a break condition, clear the LSB in BAR6 to 0.

9.3.21 Break Address Mask Register 6 (BAMR6)

BAMR6 is a 32-bit readable/writable register. BAMR6 specifies bits masked in the break address bits specified by BAR6. BAMR6 is initialized to H'00000000 by a power-on reset, but retains its previous value in sleep mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BAM6_31	BAM6_30	BAM6_29	BAM6_28	BAM6_27	BAM6_26	BAM6_25	BAM6_24	BAM6_23	BAM6_22	BAM6_21	BAM6_20	BAM6_19	BAM6_18	BAM6_17	BAM6_16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BAM6_15	BAM6_14	BAM6_13	BAM6_12	BAM6_11	BAM6_10	BAM6_9	BAM6_8	BAM6_7	BAM6_6	BAM6_5	BAM6_4	BAM6_3	BAM6_2	BAM6_1	BAM6_0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BAM6_31 to BAM6_0	All 0	R/W	Break Address Mask 6 Specify bits masked in the channel 6 break address bits specified by BAR6 (BA6_31 to BA6_0). 0: Break address bit BA6_n is included in the break condition 1: Break address bit BA6_n is masked and not included in the break condition Note: n = 31 to 0

9.3.22 Break Bus Cycle Register 6 (BBR6)

BBR6 is a 16-bit readable/writable register, which specifies (1) disabling or enabling of user break interrupts, (2) bus master of the I bus, (3) C bus cycle or I bus cycle, (4) instruction fetch or data access, (5) read or write, and (6) operand size as the break conditions of channel 6. BBR6 is initialized to H'0000 by a power-on reset, but retains its previous value in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	UBID6	-	-	CP6_ [2:0]			CD6_ [1:0]		ID6_ [1:0]		RW6_ [1:0]		SZ6_ [1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	UBID6	0	R/W	User Break Interrupt Disable 6 Disables or enables user break interrupt requests when a channel 6 break condition is satisfied. 0: User break interrupt requests enabled 1: User break interrupt requests disabled
12, 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 8	CP6_[2:0]	000	R/W	I-Bus Bus Master Select 6 Select the bus master when the bus cycle of the channel 6 break condition is the I bus cycle. However, when the C bus cycle is selected, this bit is invalidated (only the CPU cycle). xx1: CPU cycle is included in break conditions x1x: DMAC cycle is included in break conditions 1xx: A-DMAC cycle is included in break conditions

Bit	Bit Name	Initial Value	R/W	Description
7, 6	CD6_[1:0]	00	R/W	C Bus Cycle/I Bus Cycle Select 6 Select the C bus cycle or I bus cycle as the bus cycle of the channel 6 break condition. 00: Condition comparison is not performed 01: Break condition is the C bus cycle 10: Break condition is the I bus cycle 11: Break condition is the C bus cycle
5, 4	ID6_[1:0]	00	R/W	Instruction Fetch/Data Access Select 6 Select the instruction fetch cycle or data access cycle as the bus cycle of the channel 6 break condition. If the instruction fetch cycle is selected, select the C bus cycle. 00: Condition comparison is not performed 01: Break condition is the instruction fetch cycle 10: Break condition is the data access cycle 11: Break condition is the instruction fetch cycle or data access cycle
3, 2	RW6_[1:0]	00	R/W	Read/Write Select 6 Select the read cycle or write cycle as the bus cycle of the channel 6 break condition. 00: Condition comparison is not performed 01: Break condition is the read cycle 10: Break condition is the write cycle 11: Break condition is the read cycle or write cycle
1, 0	SZ6_[1:0]	00	R/W	Operand Size Select 6 Select the operand size of the bus cycle for the channel 6 break condition. 00: Break condition does not include operand size 01: Break condition is byte access 10: Break condition is word access 11: Break condition is longword access

[Legend]

x: Don't care

9.3.23 Break Address Register 7 (BAR7)

BAR7 is a 32-bit readable/writable register. BAR7 specifies the address used as a break condition in channel 7. The control bits CD7_1 and CD7_0 in the break bus cycle register 7 (BBR7) select one of the two address buses for a channel 7 break condition. BAR7 is initialized to H'00000000 by a power-on reset, but retains its previous value in sleep mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BA7_31	BA7_30	BA7_29	BA7_28	BA7_27	BA7_26	BA7_25	BA7_24	BA7_23	BA7_22	BA7_21	BA7_20	BA7_19	BA7_18	BA7_17	BA7_16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BA7_15	BA7_14	BA7_13	BA7_12	BA7_11	BA7_10	BA7_9	BA7_8	BA7_7	BA7_6	BA7_5	BA7_4	BA7_3	BA7_2	BA7_1	BA7_0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BA7_31 to BA7_0	All 0	R/W	Break Address 7 Store the CPU address bus (FAB or MAB) or IAB address specifying break conditions of channel 7. When the C bus and instruction fetch cycle are selected by BBR7, specify an FAB address in bits BA7_31 to BA7_0. When the C bus and data access cycle are selected by BBR7, specify an MAB address in bits BA7_31 to BA7_0.

Note: When setting the instruction fetch cycle as a break condition, clear the LSB in BAR7 to 0.

9.3.24 Break Address Mask Register 7 (BAMR7)

BAMR7 is a 32-bit readable/writable register. BAMR7 specifies bits masked in the break address bits specified by BAR7. BAMR7 is initialized to H'00000000 by a power-on reset, but retains its previous value in sleep mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BAM7_31	BAM7_30	BAM7_29	BAM7_28	BAM7_27	BAM7_26	BAM7_25	BAM7_24	BAM7_23	BAM7_22	BAM7_21	BAM7_20	BAM7_19	BAM7_18	BAM7_17	BAM7_16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BAM7_15	BAM7_14	BAM7_13	BAM7_12	BAM7_11	BAM7_10	BAM7_9	BAM7_8	BAM7_7	BAM7_6	BAM7_5	BAM7_4	BAM7_3	BAM7_2	BAM7_1	BAM7_0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BAM7_31 to BAM7_0	All 0	R/W	Break Address Mask 7 Specify bits masked in the channel 7 break address bits specified by BAR7 (BA7_31 to BA7_0). 0: Break address bit BA7_n is included in the break condition 1: Break address bit BA7_n is masked and not included in the break condition Note: n = 31 to 0

9.3.25 Break Bus Cycle Register 7 (BBR7)

BBR7 is a 16-bit readable/writable register, which specifies (1) disabling or enabling of user break interrupts, (2) bus master of the I bus, (3) C bus cycle or I bus cycle, (4) instruction fetch or data access, (5) read or write, and (6) operand size as the break conditions of channel 7. BBR7 is initialized to H'0000 by a power-on reset, but retains its previous value in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	UBID7	-	-	CP7_ [2:0]			CD7_ [1:0]		ID7_ [1:0]		RW7_ [1:0]		SZ7_ [1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	UBID7	0	R/W	User Break Interrupt Disable 7 Disables or enables user break interrupt requests when a channel 7 break condition is satisfied. 0: User break interrupt requests enabled 1: User break interrupt requests disabled
12, 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 8	CP7_[2:0]	000	R/W	I-Bus Bus Master Select 7 Select the bus master when the bus cycle of the channel 7 break condition is the I bus cycle. However, when the C bus cycle is selected, this bit is invalidated (only the CPU cycle). xx1: CPU cycle is included in break conditions x1x: DMAC cycle is included in break conditions 1xx: A-DMAC cycle is included in break conditions

Bit	Bit Name	Initial Value	R/W	Description
7, 6	CD7_[1:0]	00	R/W	C Bus Cycle/I Bus Cycle Select 7 Select the C bus cycle or I bus cycle as the bus cycle of the channel 7 break condition. 00: Condition comparison is not performed 01: Break condition is the C bus cycle 10: Break condition is the I bus cycle 11: Break condition is the C bus cycle
5, 4	ID7_[1:0]	00	R/W	Instruction Fetch/Data Access Select 7 Select the instruction fetch cycle or data access cycle as the bus cycle of the channel 7 break condition. If the instruction fetch cycle is selected, select the C bus cycle. 00: Condition comparison is not performed 01: Break condition is the instruction fetch cycle 10: Break condition is the data access cycle 11: Break condition is the instruction fetch cycle or data access cycle
3, 2	RW7_[1:0]	00	R/W	Read/Write Select 7 Select the read cycle or write cycle as the bus cycle of the channel 7 break condition. 00: Condition comparison is not performed 01: Break condition is the read cycle 10: Break condition is the write cycle 11: Break condition is the read cycle or write cycle
1, 0	SZ7_[1:0]	00	R/W	Operand Size Select 7 Select the operand size of the bus cycle for the channel 7 break condition. 00: Break condition does not include operand size 01: Break condition is byte access 10: Break condition is word access 11: Break condition is longword access

[Legend]

x: Don't care

9.3.26 Break Control Register 1 (BRCR1)

BRCR1 sets the following conditions:

1. Specifies whether a break of channel 4 to 7 is set before or after instruction execution.
2. Specifies whether or not the $\overline{\text{UBCTRG}}$ output is disabled when a break condition of channel 4 to 7 is satisfied.

BRCR1 is a 32-bit readable/writable register that has break condition match flags and bits for setting other break conditions. For the condition match flags of bits 15 to 8, writing 1 is invalid (previous values are retained) and writing 0 is only possible. To clear the flag, write 0 to the flag bit to be cleared and 1 to all other flag bits. BRCR1 is initialized to H'00000000 by a power-on reset, but retains its previous value in sleep mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	UTOD7	UTOD6	UTOD5	UTOD4	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SCMFC4	SCMFC5	SCMFC6	SCMFC7	SCMFD4	SCMFD5	SCMFD6	SCMFD7	PCB7	PCB6	PCB5	PCB4	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
21	UTOD7	0	R/W	$\overline{\text{UBCTRG}}$ Output Disable 7 Specifies whether or not $\overline{\text{UBCTRG}}$ output is disabled when a break condition is satisfied. 0: $\overline{\text{UBCTRG}}$ output is not disabled when a break condition for channel 7 is satisfied. 1: $\overline{\text{UBCTRG}}$ output is disabled when a break condition for channel 7 is satisfied.

Bit	Bit Name	Initial Value	R/W	Description
20	UTOD6	0	R/W	$\overline{\text{UBCTRG}}$ Output Disable 6 Specifies whether or not $\overline{\text{UBCTRG}}$ output is disabled when a break condition is satisfied. 0: $\overline{\text{UBCTRG}}$ output is not disabled when a break condition for channel 6 is satisfied. 1: $\overline{\text{UBCTRG}}$ output is disabled when a break condition for channel 6 is satisfied.
19	UTOD5	0	R/W	$\overline{\text{UBCTRG}}$ Output Disable 5 Specifies whether or not $\overline{\text{UBCTRG}}$ output is disabled when a break condition is satisfied. 0: $\overline{\text{UBCTRG}}$ output is not disabled when a break condition for channel 5 is satisfied. 1: $\overline{\text{UBCTRG}}$ output is disabled when a break condition for channel 5 is satisfied.
18	UTOD4	0	R/W	$\overline{\text{UBCTRG}}$ Output Disable 4 Specifies whether or not $\overline{\text{UBCTRG}}$ output is disabled when a break condition is satisfied. 0: $\overline{\text{UBCTRG}}$ output is not disabled when a break condition for channel 4 is satisfied. 1: $\overline{\text{UBCTRG}}$ output is disabled when a break condition for channel 4 is satisfied.
17, 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15	SCMFC4	0	R/W	C Bus Cycle Condition Match Flag 4 When the C bus cycle condition in the break conditions set for channel 4 is satisfied, this flag is set to 1. In order to clear this flag, write 0 to this bit. 0: The C bus cycle condition for channel 4 does not match 1: The C bus cycle condition for channel 4 matches

Bit	Bit Name	Initial Value	R/W	Description
14	SCMFC5	0	R/W	C Bus Cycle Condition Match Flag 5 When the C bus cycle condition in the break conditions set for channel 5 is satisfied, this flag is set to 5. In order to clear this flag, write 0 to this bit. 0: The C bus cycle condition for channel 5 does not match 1: The C bus cycle condition for channel 5 matches
13	SCMFC6	0	R/W	C Bus Cycle Condition Match Flag 6 When the C bus cycle condition in the break conditions set for channel 6 is satisfied, this flag is set to 1. In order to clear this flag, write 0 to this bit. 0: The C bus cycle condition for channel 6 does not match 1: The C bus cycle condition for channel 6 matches
12	SCMFC7	0	R/W	C Bus Cycle Condition Match Flag 7 When the C bus cycle condition in the break conditions set for channel 7 is satisfied, this flag is set to 1. In order to clear this flag, write 0 to this bit. 0: The C bus cycle condition for channel 7 does not match 1: The C bus cycle condition for channel 7 matches
11	SCMFD4	0	R/W	I Bus Cycle Condition Match Flag 4 When the I bus cycle condition in the break conditions set for channel 4 is satisfied, this flag is set to 1. In order to clear this flag, write 0 to this bit. 0: The I bus cycle condition for channel 4 does not match 1: The I bus cycle condition for channel 4 matches
10	SCMFD5	0	R/W	I Bus Cycle Condition Match Flag 5 When the I bus cycle condition in the break conditions set for channel 5 is satisfied, this flag is set to 1. In order to clear this flag, write 0 to this bit. 0: The I bus cycle condition for channel 5 does not match 1: The I bus cycle condition for channel 5 matches

Bit	Bit Name	Initial Value	R/W	Description
9	SCMFD6	0	R/W	I Bus Cycle Condition Match Flag 6 When the I bus cycle condition in the break conditions set for channel 6 is satisfied, this flag is set to 1. In order to clear this flag, write 0 to this bit. 0: The I bus cycle condition for channel 6 does not match 1: The I bus cycle condition for channel 6 matches
8	SCMFD7	0	R/W	I Bus Cycle Condition Match Flag 7 When the I bus cycle condition in the break conditions set for channel 7 is satisfied, this flag is set to 1. In order to clear this flag, write 0 to this bit. 0: The I bus cycle condition for channel 7 does not match 1: The I bus cycle condition for channel 7 matches
7	PCB7	0	R/W	PC Break Select 7 Selects the break timing of the instruction fetch cycle for channel 7 as before or after instruction execution. 0: PC break of channel 7 is generated before instruction execution 1: PC break of channel 7 is generated after instruction execution
6	PCB6	0	R/W	PC Break Select 6 Selects the break timing of the instruction fetch cycle for channel 6 as before or after instruction execution. 0: PC break of channel 6 is generated before instruction execution 1: PC break of channel 6 is generated after instruction execution
5	PCB5	0	R/W	PC Break Select 5 Selects the break timing of the instruction fetch cycle for channel 5 as before or after instruction execution. 0: PC break of channel 5 is generated before instruction execution 1: PC break of channel 5 is generated after instruction execution

Bit	Bit Name	Initial Value	R/W	Description
4	PCB4	0	R/W	PC Break Select 4 Selects the break timing of the instruction fetch cycle for channel 4 as before or after instruction execution. 0: PC break of channel 4 is generated before instruction execution 1: PC break of channel 4 is generated after instruction execution
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

9.4 Operation

9.4.1 Flow of the User Break Operation

The flow from setting of break conditions to user break exception handling is described below:

1. The break address is set in a break address register (BAR). The masked address bits are set in a break address mask register (BAMR). The bus break conditions are set in the break bus cycle register (BBR). Three control bit groups of BBR (C bus cycle/I bus cycle select, instruction fetch/data access select, and read/write select) are each set. No user break will be generated if even one of these groups is set to 00. The relevant break control conditions are set in the bits of the break control register (BRCR). Make sure to set all registers related to breaks before setting BBR, and branch after reading from the last written register and executing more than five NOP instructions. The newly written register values become valid from the instruction at the branch destination.
2. When the break conditions are satisfied, the UBC sends a user break request to the CPU, sets the C bus condition match flag (SCMFC) or I bus condition match flag (SCMFD) for the appropriate channel, and outputs a pulse to the $\overline{\text{UBCTRG}}$ pin with the width set by the CKS[1:0] bits. Setting the UBID bit in BBR to 1 enables external monitoring of the trigger output without requesting user break interrupts.
3. On receiving a user break interrupt request signal, the INTC determines its priority. Since the user break interrupt has a priority level of 15, it is accepted when the priority level set in the interrupt mask level bits (I3 to I0) of the status register (SR) is 14 or lower. If the I3 to I0 bits are set to a priority level of 15, the user break interrupt is not accepted, but the conditions are checked, and condition match flags are set if the conditions match. For details on ascertaining the priority, see section 8, Interrupt Controller (INTC).
4. Condition match flags (SCMFC and SCMFD) can be used to check which condition has been satisfied. The flag is set on a condition match and is not cleared automatically. Before using the flags again, 0 must first be written to them. The condition match flags must be cleared during the user break interrupt exception handling routine. Otherwise, an interrupt will be generated again.
5. There is a chance that the break set in channel 0 and the break set in channel 1 occur around the same time. In this case, there will be only one break request to the CPU, but these two break channel match flags may both be set.
6. When selecting the I bus as the break condition, note as follows:
   - Several bus masters, including the CPU and DMAC, are connected to the I bus. The UBC monitors bus cycles generated by the bus master specified by BBR, and determines the condition match.

- I bus cycles resulting from instruction fetches on the C bus by the CPU are defined as instruction fetch cycles on the I bus, while other bus cycles are defined as data access cycles.
- I bus cycles the DMAC or A-DMAC issues are only data access cycles.
- If a break condition is specified for the I bus, even when the condition matches in an I bus cycle resulting from an instruction executed by the CPU, at which instruction the break is to be accepted cannot be clearly defined.

9.4.2 Break on Instruction Fetch Cycle

1. When C bus/instruction fetch/read/word or longword is set in the break bus cycle register (BBR), the break condition is the FAB bus instruction fetch cycle. Whether it breaks before or after the execution of the instruction can then be selected with the PCB bit of the break control register (BRCR) for the appropriate channel. If an instruction fetch cycle is set as a break condition, clear LSB in the break address register (BAR) to 0. A break cannot be generated as long as this bit is set to 1.
2. A break for instruction fetch which is set as a break before instruction execution occurs when it is confirmed that the instruction has been fetched and will be executed. This means this function cannot be used for instructions fetched by overrun (instructions fetched at a branch or during an interrupt transition, but not to be executed). When this kind of break is set for the delay slot of a delayed branch instruction, the break is not generated until execution of the first instruction at the branch destination.

Note: If a branch does not occur at a delayed branch instruction, the subsequent instruction is not recognized as a delay slot.

3. When setting a break condition for break after instruction execution, the instruction set with the break condition is executed and then the break is generated prior to execution of the next instruction. As with pre-execution breaks, this function is not available for overrun fetch instructions. When this kind of break is set for a delayed branch instruction and its delay slot, a break is not generated until the first instruction at the branch destination.
4. If the I bus is set for a break of an instruction fetch cycle, the setting is invalidated.

9.4.3 Break on Data Access Cycle

1. If the C bus is specified as a break condition for data access break, condition comparison is performed for the logical addresses accessed by the executed instructions, and a break occurs if the condition is satisfied. If the I bus is specified as a break condition, condition comparison is performed for the physical addresses of the data access cycles that are issued by the bus master specified by the bits to select the bus master of the I bus, and a break occurs if the condition is satisfied. For details on the CPU bus cycles issued on the I bus, see 6 in section 9.4.1, Flow of the User Break Operation.
2. The relationship between the data access cycle address and the comparison condition for each operand size is listed in table 9.3.

Table 9.3 Data Access Cycle Addresses and Operand Size Comparison Conditions

Access Size	Address Compared
Longword	Compares break address register bits 31 to 2 to address bus bits 31 to 2
Word	Compares break address register bits 31 to 1 to address bus bits 31 to 1
Byte	Compares break address register bits 31 to 0 to address bus bits 31 to 0

   This means that when address H'00001003 is set in the break address register (BAR), for example, the bus cycle in which the break condition is satisfied is as follows (where other conditions are met).

   Longword access at H'00001000

   Word access at H'00001002

   Byte access at H'00001003

3. If the data access cycle is selected, the instruction at which the break will occur cannot be determined.

9.4.4 Value of Saved Program Counter

When a break occurs, the address of the instruction from where execution is to be resumed is saved to the stack, and the exception handling state is entered. If the C bus (FAB)/instruction fetch cycle is specified as a break condition, the instruction at which the break should occur can be uniquely determined. If the C bus/data access cycle or I bus/data access cycle is specified as a break condition, the instruction at which the break should occur cannot be uniquely determined.

1. When C bus (FAB)/instruction fetch (before instruction execution) is specified as a break condition:

   The address of the instruction that matched the break condition is saved to the stack. The instruction that matched the condition is not executed, and the break occurs before it. However when a delay slot instruction matches the condition, the instruction is executed, and the branch destination address is saved to the stack.
2. When C bus (FAB)/instruction fetch (after instruction execution) is specified as a break condition:

   The address of the instruction following the instruction that matched the break condition is saved to the stack. The instruction that matches the condition is executed, and the break occurs before the next instruction is executed. However when a delayed branch instruction or delay slot matches the condition, the instruction is executed, and the branch destination address is saved to the stack.
3. When C bus/data access or I bus/data access is specified as a break condition:

   The address after executing several instructions of the instruction that matched the break condition is saved to the stack.

9.4.5 Usage Examples

(1) Break Condition Specified for C Bus Instruction Fetch Cycle

(Example 1-1)

- Register specifications

  BAR0 = H'00000404, BAMR0 = H'00000000, BBR0 = H'0054, BAR1 = H'00008010, BAMR1 = H'00000006, BBR1 = H'0054, BRCR = H'00000010

  <Channel 0>

  Address: H'00000404, Address mask: H'00000000

  Bus cycle: C bus/instruction fetch (after instruction execution)/read (operand size is not included in the condition)

  <Channel 1>

  Address: H'00008010, Address mask: H'00000006

  Data: H'00000000, Data mask: H'00000000

  Bus cycle: C bus/instruction fetch (before instruction execution)/read (operand size is not included in the condition)

  A user break occurs after an instruction of address H'00000404 is executed or before instructions of addresses H'00008010 to H'00008016 are executed.

(Example 1-2)

- Register specifications

  BAR0 = H'00027128, BAMR0 = H'00000000, BBR0 = H'005A, BAR1= H'00031415, BAMR1 = H'00000000, BBR1 = H'0054, BRCR = H'00000000

  <Channel 0>

  Address: H'00027128, Address mask: H'00000000

  Bus cycle: C bus/instruction fetch (before instruction execution)/write/word

  <Channel 1>

  Address: H'00031415, Address mask: H'00000000

  Data: H'00000000, Data mask: H'00000000

  Bus cycle: C bus/instruction fetch (before instruction execution)/read (operand size is not included in the condition)

  On channel 0, a user break does not occur since instruction fetch is not a write cycle. On channel 1, a user break does not occur since instruction fetch is performed for an even address.

(Example 1-3)

- Register specifications

  BAR0 = H'00008404, BAMR0 = H'00000FFF, BBR0 = H'0054, BAR1= H'00008010, BAMR1 = H'00000006, BBR1 = H'0054, BRCR = H'00000010

  <Channel 0>

  Address: H'00008404, Address mask: H'00000FFF

  Bus cycle: C bus/instruction fetch (after instruction execution)/read (operand size is not included in the condition)

  <Channel 1>

  Address: H'00008010, Address mask: H'00000006

  Data: H'00000000, Data mask: H'00000000

  Bus cycle: C bus/instruction fetch (before instruction execution)/read (operand size is not included in the condition)

  A user break occurs after an instruction with addresses H'00008000 to H'00008FFE is executed or before an instruction with addresses H'00008010 to H'00008016 are executed.

(2) Break Condition Specified for C Bus Data Access Cycle

(Example 2-1)

- Register specifications

  BAR0 = H'00123456, BAMR0 = H'00000000, BBR0 = H'0064, BAR1= H'000ABCDE, BAMR1 = H'000000FF, BBR1 = H'006A, BRCR = H'00000000

  <Channel 0>

  Address: H'00123456, Address mask: H'00000000

  Bus cycle: C bus/data access/read (operand size is not included in the condition)

  <Channel 1>

  Address: H'000ABCDE, Address mask: H'000000FF

  Data: H'0000A512, Data mask: H'00000000

  Bus cycle: C bus/data access/write/word

  On channel 0, a user break occurs with longword read from address H'00123456, word read from address H'00123456, or byte read from address H'00123456. On channel 1, a user break occurs when word data is written in addresses H'000ABC00 to H'000ABCFE.

(3) Break Condition Specified for I Bus Data Access Cycle

(Example 3-1)

- Register specifications

  BAR0 = H'00314156, BAMR0 = H'00000000, BBR0 = H'0094, BAR1= H'00055555, BAMR1 = H'00000000, BBR1 = H'02A9, BRCR = H'00000000

  <Channel 0>

  Address: H'00314156, Address mask: H'00000000

  Bus cycle: I bus/instruction fetch/read (operand size is not included in the condition)

  <Channel 1>

  Address: H'00055555, Address mask: H'00000000

  Data: H'00000078, Data mask: H'0000000F

  Bus cycle: I bus/data access/write/byte

  On channel 0, the setting of I bus/instruction fetch is ignored.

  On channel 1, a user break occurs when the DMAC writes byte data in address H'00055555 on the I bus (write by the CPU does not generate a user break).

9.5 Usage Notes

1. The CPU can read from or write to the UBC registers via the I bus. Accordingly, during the period from executing an instruction to rewrite the UBC register till the new value is actually rewritten, the desired break may not occur. In order to know the timing when the UBC register is changed, execute five or more NOP instructions after reading from the last written register. Instructions after then are valid for the newly written register value.
2. The UBC cannot monitor access to the C bus and I bus cycles in the same channel.
3. When a user break and another exception occur at the same instruction, which has higher priority is determined according to the priority levels defined in table 7.1 in section 7, Exception Handling. If an exception with higher priority occurs, the user break is not generated.
4. Note the following when a break occurs in a delay slot.

   If a pre-execution break is set at a delay slot instruction, the break occurs immediately before execution of the branch destination.
5. User breaks are disabled during UBC module standby mode. Do not read from or write to the UBC registers during UBC module standby mode; the values are not guaranteed.
6. Do not set an address within an interrupt exception handling routine whose interrupt priority level is at least 15 (including user break interrupts) as a break address.
7. Do not set a break after instruction execution for the SLEEP instruction or for the delayed branch instruction where the SLEEP instruction is placed at its delay slot.
8. When setting a break for a 32-bit instruction, set the address where the upper 16 bits are placed. If the address of the lower 16 bits is set and a break before instruction execution is set as a break condition, the break is handled as a break after instruction execution.
9. Do not place a pre-execution break for the instruction immediately after a DIVU (or DIVS) instruction. Otherwise the break will be still taken even though an exception or interrupt occurs during the DIVU (or DIVS) instruction execution.

Section 10 Bus State Controller (BSC)

The bus state controller (BSC) outputs control signals for external memory and external devices that are connected to the external address space. BSC functions enable this LSI to connect directly with SRAM, ROM, and other memories such as SRAM with byte selection, and external devices.

10.1 Features

1. External address space
    - A maximum 4-Mbyte linear space access for each of spaces CS0 to CS3.
    - Can select the data bus width (8 or 16 bits) for each address space.
    - Controls insertion of wait cycles for each address space.
    - Controls insertion of wait cycles for each read access and write access.
    - Can set independent idle cycles during the continuous access for five cases: read-write (in same space/different spaces), read-read (in same space/different spaces), the first cycle is a write access.

Figure 10.1 shows a block diagram of the BSC.

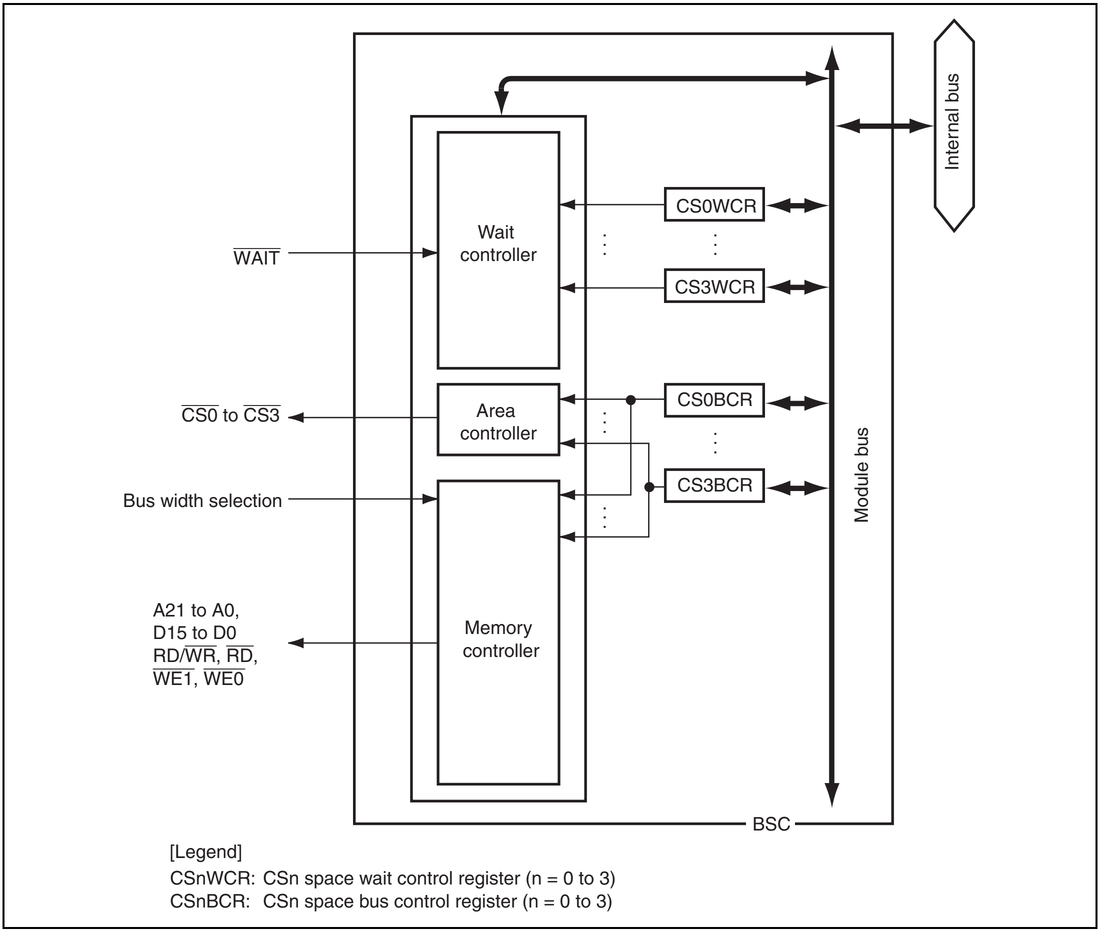

Internal bus
Module bus
BSC
Wait controller
Area controller
Memory controller
CS0WCR
CS3WCR
CS0BCR
CS3BCR
$\overline{\text{WAIT}}$
$\overline{\text{CS0}}$ to $\overline{\text{CS3}}$
Bus width selection
A21 to A0,
D15 to D0
RD/$\overline{\text{WR}}$, $\overline{\text{RD}}$,
$\overline{\text{WE1}}$, $\overline{\text{WE0}}$
[Legend]
CSnWCR: CSn space wait control register (n = 0 to 3)
CSnBCR: CSn space bus control register (n = 0 to 3)

Figure 10.1 Block Diagram of BSC

10.2 Input/Output Pins

Table 10.1 shows the pin configuration of the BSC.

Table 10.1 Pin Configuration

Name	I/O	Function
A21 to A0	Output	External address bus
D15 to D0	I/O	External data bus
$\overline{\text{CS0}}$ to $\overline{\text{CS3}}$	Output	Chip select
RD/$\overline{\text{WR}}$	Output	Read/write Can be connected to $\overline{\text{WE}}$ pins when SRAM with byte selection is connected.
$\overline{\text{RD}}$	Output	Read pulse signal (read data output enable signal)
$\overline{\text{WE1}}$	Output	Indicates that D15 to D8 are being written to. Indicates the byte select signal when a SRAM with byte selection is connected.
$\overline{\text{WE0}}$	Output	Indicates that D7 to D0 are being written to. Indicates the byte select signal when a SRAM with byte selection is connected.
$\overline{\text{WAIT}}$	Input	External wait input
Bus width selection	Input	Select the initial bus width of address spaces CS0 to CS3.

10.3 Area Overview

10.3.1 Address Map

In the architecture, this LSI has a 32-bit address space, which is divided into on-chip address spaces (on-chip RAM, on-chip I/O registers, and reserved areas), external address spaces, and on-chip ROM spaces according to the upper six bits of the address.

Any memory space of CS0 to CS3 can be activated by SRAM interface or SRAM interface with byte selection. However, the space CS0 in on-chip ROM disabled mode should be activated by SRAM interface.

The address map for the external address space is listed in tables 10.2 and 10.3.

Table 10.2 Address Space of the On-Chip ROM Enabled Mode (Exclusive of Single-Chip Mode)

Address	Type of address space	Size
H'0000 0000 to H'003F FFFF (H'0000 0000 to H'0000 7FFF)*[1]	On-chip ROM (read from user MAT) (on-chip ROM (read from user boot MAT))	4 Mbytes (32 Kbytes)
H'0040 0000 to H'0040 1FFF	Reserved area	
H'0040 2000 to H'0040 3FFF	FCU firmware area	8 Kbytes
H'0040 4000 to H'01FF FFFF	Reserved area	
H'0200 0000 to H'023F FFFF	External address space (CS0)	4 Mbytes
H'0240 0000 to H'03FF FFFF	CS0 shadow space	28 Mbytes
H'0400 0000 to H'043F FFFF	External address space (CS1)	4 Mbytes
H'0440 0000 to H'07FF FFFF	CS1 shadow space	60 Mbytes
H'0800 0000 to H'083F FFFF	External address space (CS2)	4 Mbytes
H'0840 0000 to H'0BFF FFFF	CS2 shadow space	60 Mbytes
H'0C00 0000 to H'0C3F FFFF	External address space (CS3)	4 Mbytes
H'0C40 0000 to H'0FFF FFFF	CS3 shadow space	60 Mbytes
H'1000 0000 to H'800F FFFF	Reserved area	
H'8010 0000 to H'8011 FFFF	EEPROM (read/write)*[2]	128 Kbytes
H'8012 0000 to H'807F FFFF	Reserved area	
H'8080 0000 to H'80BB FFFF (H'8080 0000 to H'8080 7FFF)*[1]	On-chip ROM (write to user MAT) (on-chip ROM (write to user boot MAT))	4 Mbytes (32 Kbytes)

Address	Type of address space	Size
H'80C0 0000 to H'80FF 7FFF	Reserved area	
H'80FF 8000 to H'80FF 9FFF	FCU RAM area	8 Kbytes
H'80FF A000 to H'FFF7 FFFF	Reserved area	
H'FFF8 0000 to H'FFFB FFFF	On-chip RAM*[3]	256 Kbytes
H'FFFC 0000 to H'FFFF FFFF	Internal I/O register	256 Kbytes (Max.)

Notes: 1. When the user boot MAT of the on-chip ROM is selected. See section 26, ROM.

2. The EEPROM area includes the lot trace information. See section 28, EEPROM.

Table 10.3 Address Space of the On-Chip ROM Disabled Mode

Address	Type of address space	Size
H'0000 0000 to H'003F FFFF	External address space (CS0)	4 Mbytes
H'0040 0000 to H'03FF FFFF	CS0 shadow space	60 Mbytes
H'0400 0000 to H'043F FFFF	External address space (CS1)	4 Mbytes
H'0440 0000 to H'07FF FFFF	CS1 shadow space	60 Mbytes
H'0800 0000 to H'083F FFFF	External address space (CS2)	4 Mbytes
H'0840 0000 to H'0BFF FFFF	CS2 shadow space	60 Mbytes
H'0C00 0000 to H'0C3F FFFF	External address space (CS3)	4 Mbytes
H'0C40 0000 to H'0FFF FFFF	CS3 shadow space	60 Mbytes
H'1000 0000 to H'FFF7 FFFF	Reserved area	
H'FFF8 0000 to H'FFFB FFFF	On-chip RAM	256 Kbytes
H'FFFC 0000 to H'FFFF FFFF	Internal I/O register	256 Kbytes (Max.)

Table 10.4 Address Space of the Single-Chip Mode

Address	Type of address space	Size
H'0000 0000 to H'003F FFFF (H'0000 0000 to H'0000 7FFF)*[1]	On-chip ROM (read from user MAT) (on-chip ROM (read from user boot MAT))	4 Mbytes (32 Kbytes)
H'0040 0000 to H'0040 1FFF	Reserved area	
H'0040 2000 to H'0040 3FFF	FCU firmware area	8 Kbytes
H'0040 4000 to H'800F FFFF	Reserved area	
H'8010 0000 to H'8011 FFFF	EEPROM (read/write)*[2]	128 Kbytes
H'8012 0000 to H'807F FFFF	Reserved area	
H'8080 0000 to H'80BB FFFF (H'8080 0000 to H'8080 7FFF)*[1]	On-chip ROM (write to user MAT) (on-chip ROM (write to user boot MAT))	4 Mbytes (32 Kbytes)
H'80C0 0000 to H'80FF 7FFF	Reserved area	
H'80FF 8000 to H'80FF 9FFF	FCU RAM area	8 Kbytes
H'80FF A000 to H'FFF7 FFFF	Reserved area	
H'FFF8 0000 to H'FFFB FFFF	On-chip RAM*[3]	256 Kbytes
H'FFFC 0000 to H'FFFF FFFF	Internal I/O register	256 Kbytes (Max.)

Notes: 1. When the user boot MAT of the on-chip ROM is selected. See section 26, ROM.

2. The EEPROM area includes the lot trace information. See section 28, EEPROM.

3. For the on-chip RAM space, access the addresses shown in section 5, Address Space. For the on-chip I/O register space, access the addresses shown in section 33, List of Registers. Do not access addresses which are not described in these sections. Otherwise, the correct operation cannot be guaranteed.

10.3.2 Operating Modes and Data Bus Width in CS0 Space

This LSI can specify the following operating modes at a power-on reset by using the mode setting pins.

(1) Single Chip Mode/External Address Space Access Enabled Mode

In single chip mode, the external address spaces are not accessed. The addresses, data, and control pins to be used in external address space access enabled mode can be used as ports.

(2) On-Chip ROM Enabled Mode/On-Chip ROM Disabled Mode

In on-chip ROM enabled mode, first half of the CS0 is allocated in the on-chip ROM. Thus, after a power-on reset, the LSI can be activated by the on-chip ROM program.

In on-chip ROM disabled mode, the LSI is activated by the program stored in the external memory allocated in the CS0. In this case, the external memory in the CS0 functions as the SRAM interface, therefore, the SRAM with byte selection cannot be connected.

To use the RD/$\overline{\text{WR}}$ signal or $\overline{\text{CS1}}$ to $\overline{\text{CS3}}$ signals, these pins must be set by the pin function controller. For details, see section 23, Pin Function Controller (PFC). Execute nothing but accessing CS0 space before a program completes pin function settings.

(3) Data Bus Width of the Space CS0

In on-chip ROM disabled mode, the data bus width of CS0 area can be set to 8 bits or 16 bits. The data bus widths of CS1 to CS3 areas are specified through register settings. In on-chip ROM enabled mode, all of the data bus widths of the spaces CS0 to CS3 are specified through register settings.

For details on mode settings, see section 3, Operating Modes.

10.4 Register Descriptions

The BSC has the following registers.

For addresses and access sizes of these registers, see section 33, List of Registers.

In on-chip ROM disabled mode, do not access spaces other than CS0 area until settings of the connected memory interface are completed.

Table 10.5 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
CSn space bus control register	CSnBCR	R/W	H'36DB0400* H'36DB0200*	H'FFFC0004 to H'FFFC0010	32
CSn space wait control register	CSnWCR	R/W	H'00000500	H'FFFC0028 to H'FFFC0034	32

Note: * This initial value is different between in on-chip ROM enabled mode and on-chip ROM disabled mode. In on-chip ROM disabled mode, the initial value varies depending on the settings of the mode pins. For details, see section 10.4.1, CSn Space Bus Control Register (CSnBCR) (n = 0 to 3).

10.4.1 CSn Space Bus Control Register (CSnBCR) (n = 0 to 3)

CSnBCR is a 32-bit readable/writable register that specifies the function of each area, the number of idle cycles between bus cycles, and the bus width. In on-chip ROM enabled mode, CSnBCR is initialized to H'36DB0400 at a power-on reset. In on-chip ROM disabled mode, it is initialized to H'36DB0400 (16-bit bus width) or H'36DB0200 (8-bit bus width) depending on the settings of the mode pins.

Do not access external memory other than CS0 before CSnBCR initial setting is completed.

Idle cycles may be inserted even when they are not specified. For details, see section 10.5.6, Wait between Access Cycles.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	IWW[2:0]			IWRWD[2:0]			IWRWS[2:0]			IWRRD[2:0]			IWRRS[2:0]		
Initial value:	0	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	TYPE[1:0]		-	BSZ[1:0]		-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	1/0	0/1	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R/W	R/W	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
30 to 28	IWW[2:0]	011	R/W	Idle Cycles between Write-Read Cycles and Write-Write Cycles These bits specify the number of idle cycles to be inserted after the access to a memory that is connected to the space. The target access cycles are the write-read cycle and write-write cycle. 000: No idle cycle inserted (setting prohibited when two-times multiplication has been set for the peripheral clock) 001: 1 idle cycle inserted (setting prohibited when two-times multiplication has been set for the peripheral clock) 010: 2 idle cycles inserted 011: 4 idle cycles inserted 100: 6 idle cycles inserted 101: 8 idle cycles inserted 110: 10 idle cycles inserted 111: 12 idle cycles inserted

Bit	Bit Name	Initial Value	R/W	Description
27 to 25	IWRWD[2:0]	011	R/W	Idle Cycles for Another Space Read-Write Specify the number of idle cycles to be inserted after the access to a memory that is connected to the space. The target access cycle is a read-write one in which continuous access cycles switch between different spaces. 000: No idle cycle inserted (setting prohibited when two-times multiplication has been set for the peripheral clock) 001: 1 idle cycle inserted (setting prohibited when two-times multiplication has been set for the peripheral clock) 010: 2 idle cycles inserted 011: 4 idle cycles inserted 100: 6 idle cycles inserted 101: 8 idle cycles inserted 110: 10 idle cycles inserted 111: 12 idle cycles inserted
24 to 22	IWRWS[2:0]	011	R/W	Idle Cycles for Read-Write in the Same Space Specify the number of idle cycles to be inserted after the access to a memory that is connected to the space. The target cycle is a read-write cycle of which continuous access cycles are for the same space. 000: No idle cycle inserted (setting prohibited when two-times multiplication has been set for the peripheral clock) 001: 1 idle cycle inserted (setting prohibited when two-times multiplication has been set for the peripheral clock) 010: 2 idle cycles inserted 011: 4 idle cycles inserted 100: 6 idle cycles inserted 101: 8 idle cycles inserted 110: 10 idle cycles inserted 111: 12 idle cycles inserted

Bit	Bit Name	Initial Value	R/W	Description
21 to 19	IWRRD[2:0]	011	R/W	Idle Cycles for Read-Read in Another Space Specify the number of idle cycles to be inserted after the access to a memory that is connected to the space. The target cycle is a read-read cycle of which continuous access cycles switch between different spaces. 000: No idle cycle inserted (setting prohibited when two-times multiplication has been set for the peripheral clock) 001: 1 idle cycle inserted (setting prohibited when two-times multiplication has been set for the peripheral clock) 010: 2 idle cycles inserted 011: 4 idle cycles inserted 100: 6 idle cycles inserted 101: 8 idle cycles inserted 110: 10 idle cycles inserted 111: 12 idle cycles inserted
18 to 16	IWRRS[2:0]	011	R/W	Idle Cycles for Read-Read in the Same Space Specify the number of idle cycles to be inserted after the access to a memory that is connected to the space. The target cycle is a read-read cycle of which continuous access cycles are for the same space. 000: No idle cycle inserted (setting prohibited when two-times multiplication has been set for the peripheral clock) 001: 1 idle cycle inserted (setting prohibited when two-times multiplication has been set for the peripheral clock) 010: 2 idle cycles inserted 011: 4 idle cycles inserted 100: 6 idle cycles inserted 101: 8 idle cycles inserted 110: 10 idle cycles inserted 111: 12 idle cycles inserted

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13, 12	TYPE[1:0]	00	R/W	Specify the type of memory connected to a space. 00: Normal space (SRAM) 01: Reserved (setting prohibited) 10: Reserved (setting prohibited) 11: SRAM with byte selection
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10, 9	BSZ[1:0]	10/01	R/W	Data Bus Width Specification Specify the data bus widths of spaces. 00: Reserved (setting prohibited) 01: 8-bit size 10: 16-bit size 11: Reserved (setting prohibited) Note: In on-chip RAM disabled mode, the initial data bus widths set in the CS0 to CS3 are specified through the mode pins settings. The mode setting pin data sampled at a power-on reset is reflected to the BSZ1 and BSZ0 bits. In on-chip ROM disabled mode, writing to the BSZ1 and BSZ0 bits in CS0BCR are ignored, but the bus width set in CS1BCR to CS3BCR can be modified. In on-chip ROM enabled mode, the initial data bus widths set in CS0 to CS3 are specified as 16 bits. In this case, the bus width set in CS0BCR to CS3BCR can be modified.
8 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

10.4.2 CSn Space Wait Control Register (CSnWCR) (n = 0 to 3)

CSnWCR specifies various wait cycles for memory access. Specify CSnBCR first, and then specify CSnWCR.

CSnWCR is initialized to H'00000500 by a power-on reset.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	BAS	-	WW[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	SW[1:0]		WR[3:0]				WM	-	-	-	-	HW[1:0]	
Initial value:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20	BAS	0	R/W	SRAM with Byte Selection Byte Access Select Specifies the $\overline{\text{WEn}}$ and RD/$\overline{\text{WR}}$ signal timing when the SRAM interface with byte selection is used. 0: Asserts the $\overline{\text{WEn}}$ signal at the read/write timing and asserts the RD/$\overline{\text{WR}}$ signal during the write access cycle. 1: Asserts the $\overline{\text{WEn}}$ signal during the read/write access cycle and asserts the RD/$\overline{\text{WR}}$ signal at the write timing.
19	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
18 to 16	WW[2:0]	000	R/W	Number of Write Access Wait Cycles Specify the number of cycles that are necessary for write access. 000: The same cycles as WR[3:0] setting (number of read access wait cycles) 001: No cycle (setting prohibited when two-times multiplication has been set for the peripheral clock) 010: 1 cycle 011: 2 cycles 100: 3 cycles 101: 4 cycles 110: 5 cycles 111: 6 cycles
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
12, 11	SW[1:0]	00	R/W	When normal SRAM connection or SRAM with byte selection connection is used with BAS = 0: • Number of Delay Cycles from Address, $\overline{\text{CSn}}$ Assertion to $\overline{\text{RD}}$, $\overline{\text{WEn}}$ Assertion Specify the number of delay cycles from address and $\overline{\text{CSn}}$ assertion to $\overline{\text{RD}}$ and $\overline{\text{WEn}}$ assertion. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles For SRAM with byte selection connection with BAS = 1: • Number of Delay Cycles from Address, $\overline{\text{CSn}}$ Assertion to $\overline{\text{RD}}$ Assertion Specify the number of delay cycles from address and $\overline{\text{CSn}}$ assertion to $\overline{\text{RD}}$ assertion. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles • Number of Delay Cycles from Address, $\overline{\text{CSn}}$ Assertion to RD/$\overline{\text{WR}}$ Assertion Specify the number of delay cycles from address and $\overline{\text{CSn}}$ assertion to RD/$\overline{\text{WR}}$ assertion. 00: 1 cycle 01: 2 cycles 10: 3 cycles 11: 4 cycles

Bit	Bit Name	Initial Value	R/W	Description
10 to 7	WR[3:0]	1010	R/W	Number of Read Access Wait Cycles Specify the number of cycles that are necessary for read access. 0000: No cycle (setting prohibited when two-times multiplication has been set for the peripheral clock) 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)
6	WM	0	R/W	External Wait Mask Specification Specifies whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycle is 0. 0: External wait input is valid 1: External wait input is ignored
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
1, 0	HW[1:0]	00	R/W	When normal SRAM connection or SRAM with byte selection connection is used with BAS = 0: • Number of Delay Cycles from $\overline{\text{RD}}$, $\overline{\text{WEn}}$ Negation to Address, $\overline{\text{CSn}}$ Negation Specify the number of delay cycles from $\overline{\text{RD}}$ and $\overline{\text{WEn}}$ negation to address and $\overline{\text{CSn}}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles For SRAM with byte selection connection with BAS = 1: • Number of Delay Cycles from $\overline{\text{RD}}$ Negation to Address, $\overline{\text{CSn}}$ Negation Specify the number of delay cycles from $\overline{\text{RD}}$ negation to address and $\overline{\text{CSn}}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles • Number of Delay Cycles from RD/$\overline{\text{WR}}$ Negation to Address, $\overline{\text{CSn}}$ Negation Specify the number of delay cycles from RD/$\overline{\text{WR}}$ negation to address and $\overline{\text{CSn}}$ negation. 00: 1 cycle 01: 2 cycles 10: 3 cycles 11: 4 cycles

10.5 Operation

10.5.1 Endian/Access Size and Data Alignment

This LSI supports big endian, in which the 0 address is the most significant byte (MSB) in the byte data. Two data bus widths (8 bits and 16 bits) are available.

Data alignment is performed in accordance with the data bus width of the CS space. This also means that when longword data is read from a byte-width CS space, the read operation must be done four times. In this LSI, data alignment and conversion of data length is performed automatically between the respective CS spaces.

Tables 10.6 and 10.7 show the relationship between device data width and access unit.

Table 10.6 16-Bit External Device Access and Data Alignment

		Data Bus		Strobe Signals	
Operation		**D15 to D8**	**D7 to D0**	$\overline{\text{WE1}}$	$\overline{\text{WE0}}$
Byte access at 0		Data 7 to 0	—	Assert	—
Byte access at 1		—	Data 7 to 0	—	Assert
Byte access at 2		Data 7 to 0	—	Assert	—
Byte access at 3		—	Data 7 to 0	—	Assert
Word access at 0		Data 15 to 8	Data 7 to 0	Assert	Assert
Word access at 2		Data 15 to 8	Data 7 to 0	Assert	Assert
Longword access at 0	1st time at 0	Data 31 to 24	Data 23 to 16	Assert	Assert
	2nd time at 2	Data 15 to 8	Data 7 to 0	Assert	Assert

Table 10.7 8-Bit External Device Access and Data Alignment

		Data Bus		Strobe Signals	
Operation		**D15 to D8**	**D7 to D0**	$\overline{\text{WE1}}$	$\overline{\text{WE0}}$
Byte access at 0		—	Data 7 to 0	—	Assert
Byte access at 1		—	Data 7 to 0	—	Assert
Byte access at 2		—	Data 7 to 0	—	Assert
Byte access at 3		—	Data 7 to 0	—	Assert
Word access at 0	1st time at 0	—	Data 15 to 8	—	Assert
	2nd time at 1	—	Data 7 to 0	—	Assert
Word access at 2	1st time at 2	—	Data 15 to 8	—	Assert
	2nd time at 3	—	Data 7 to 0	—	Assert
Longword access at 0	1st time at 0	—	Data 31 to 24	—	Assert
	2nd time at 1	—	Data 23 to 16	—	Assert
	3rd time at 2	—	Data 15 to 8	—	Assert
	4th time at 3	—	Data 7 to 0	—	Assert

10.5.2 External Space Interface

(1) Basic Timing for One-time Multiplication Peripheral Clock

For access to an external space, this LSI uses strobe signal output in consideration of the fact that mainly static RAM or ROM will be directly connected. Figure 10.2 shows the basic timings. A no-wait normal access is completed in two cycles.

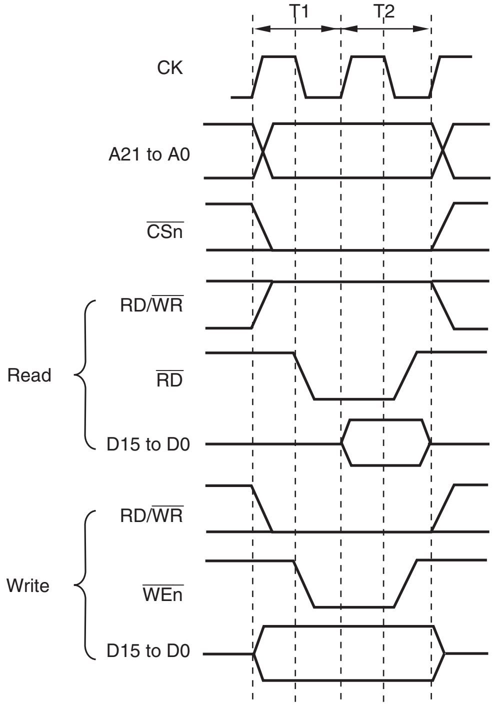


Figure 10.2 Basic Access Timing (Access Wait 0)

There is no access size specification when reading. The correct access start address is output in the least significant bit of the address, but since there is no access size specification, 16 bits are always read in case of a 16-bit apace, and 8 bits in case of an 8-bit space. When writing, only the $\overline{\text{WEn}}$ signal for the byte to be written is asserted.

It is necessary to output the data that has been read using $\overline{\text{RD}}$ when a buffer is established in the data bus. The RD/$\overline{\text{WR}}$ signal is in a read state (high output) when no access has been carried out. Therefore, care must be taken when controlling the external data buffer, to avoid collision.

Figures 10.3 and 10.4 show the basic timings of continuous access. If the WM bit in CSnWCR is cleared to 0, a Tnop cycle is inserted after the CSn space access to evaluate the external wait (figure 10.3). If the WM bit in CSnWCR is set to 1, external waits are ignored and no Tnop cycle is inserted (figure 10.4).

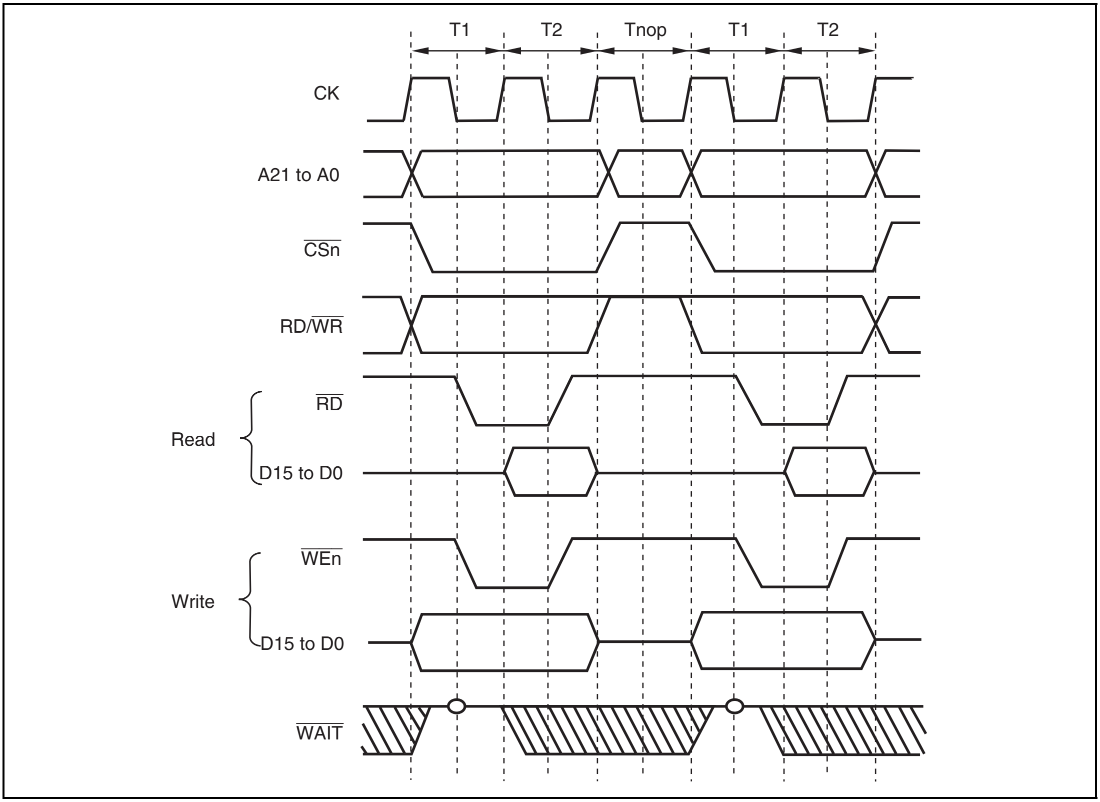


Figure 10.3 Continuous Access 1
Bus Width = 16 Bits, Longword Access, CSnWCR.WM Bit = 0
(Access Wait = 0, Cycle Wait = 0)

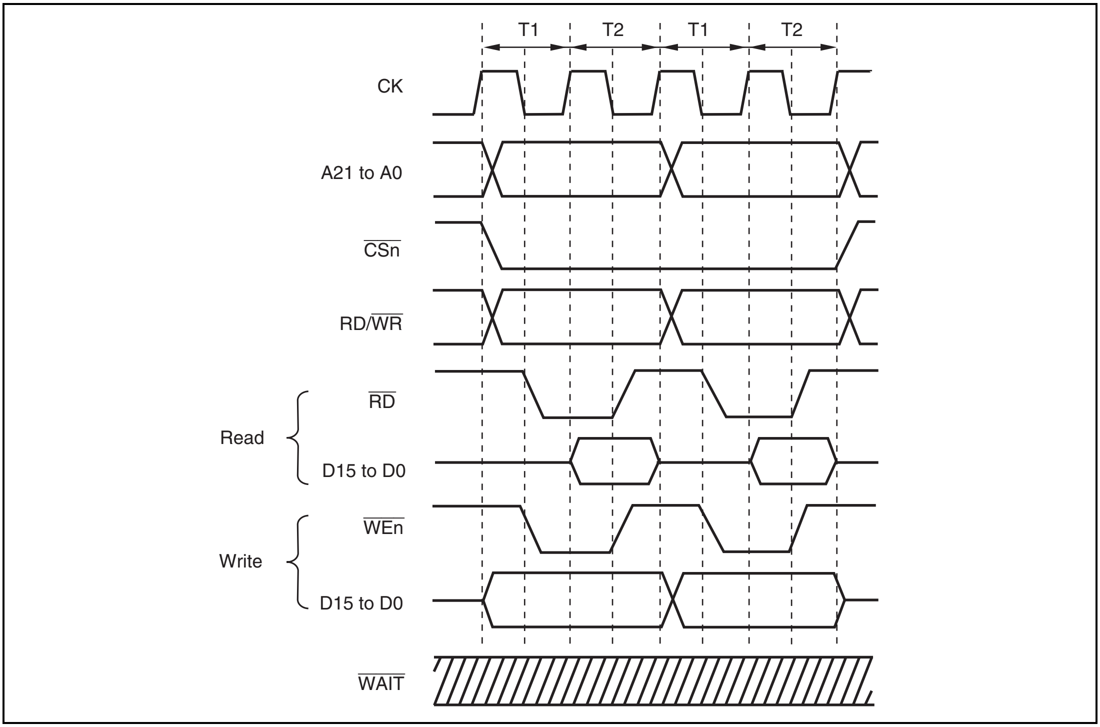


Figure 10.4 Continuous Access 2
Bus Width = 16 Bits, Longword Access, CSnWCR.WM Bit = 1
(Access Wait = 0, Cycle Wait = 0)

(2) Basic Timing for Two-Times Multiplied Peripheral Clock

When two-times multiplication has been set for the peripheral clock, CSnBCR and CSnWCR must be set so that more than one cycle and two cycles are inserted as Tw and Tid, respectively. Figure 10.5 shows the basic timing for two-times multiplied peripheral clock. Figure 10.6 shows a continuous access to normal space for two-times multiplied peripheral clock. Setting the CSnWCR.WR[3:0] bits to 0001 inserts one cycle as Tw (Figure 10.5). Setting the IWW[2:0], IWRWD[2:0], IWRWS[2:0], IWRRD[2:0], and IWRRS[2:0] bits in CSnBCR to 010 inserts two cycles as Tid after access to the CSn space (Figure 10.6). Even when the CSnWCR.WM bit is 0, no Tnop is inserted (see section 10.5.6, Wait between Access Cycles).

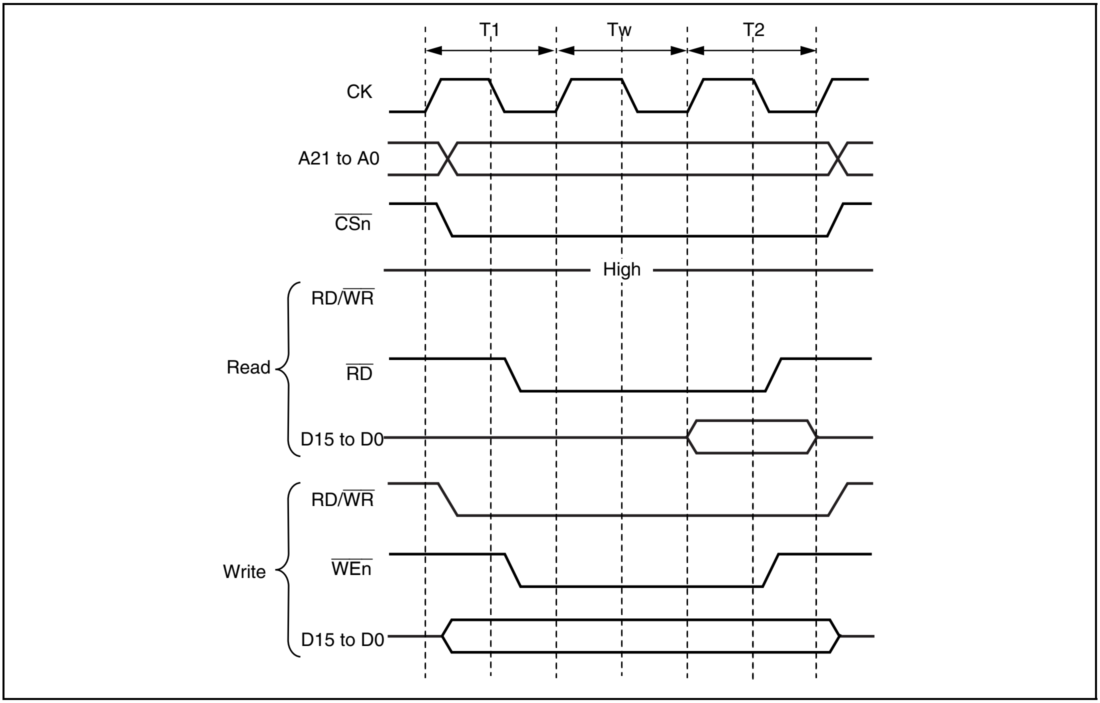


Figure 10.5 Basic Access for Two-Times Multiplied Peripheral Clock (Access Wait 1)

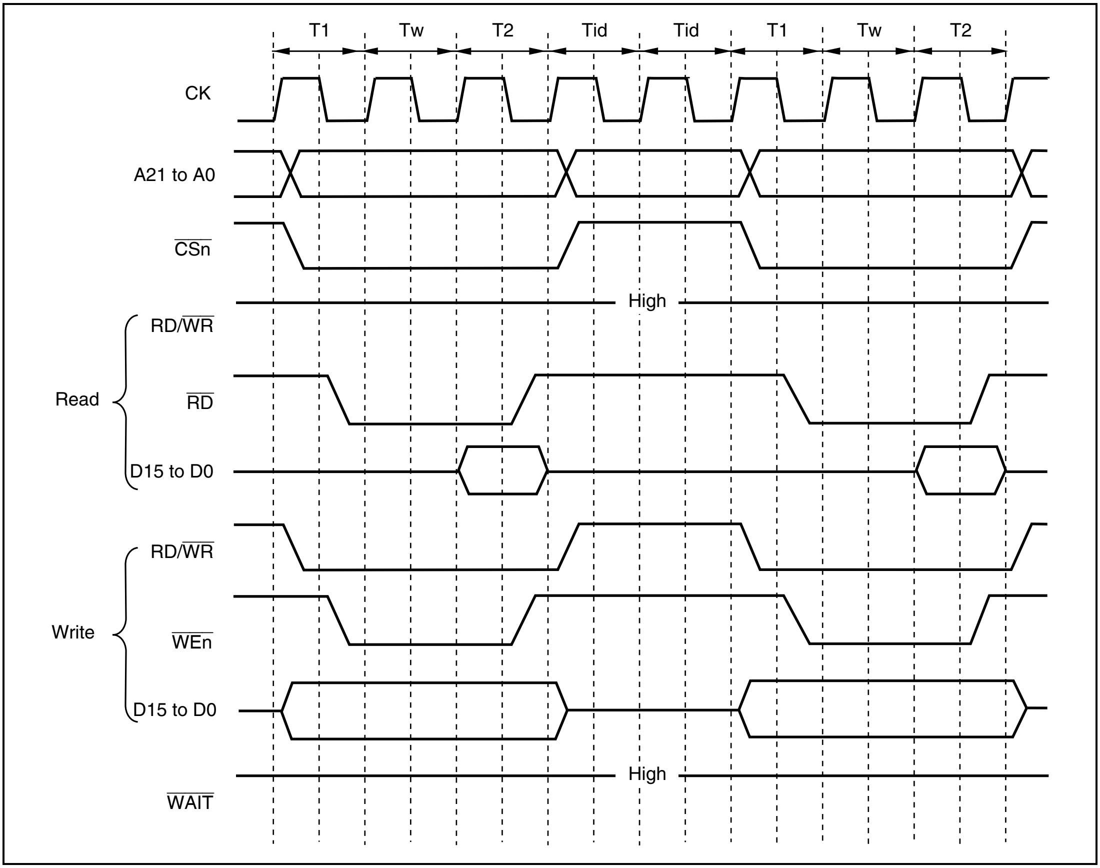


Figure 10.6 Continuous Access for Two-Times Multiplied Peripheral Clock Bus Width = 16 Bits, Longword Access (Access Wait = 1, Cycle Wait = 2)

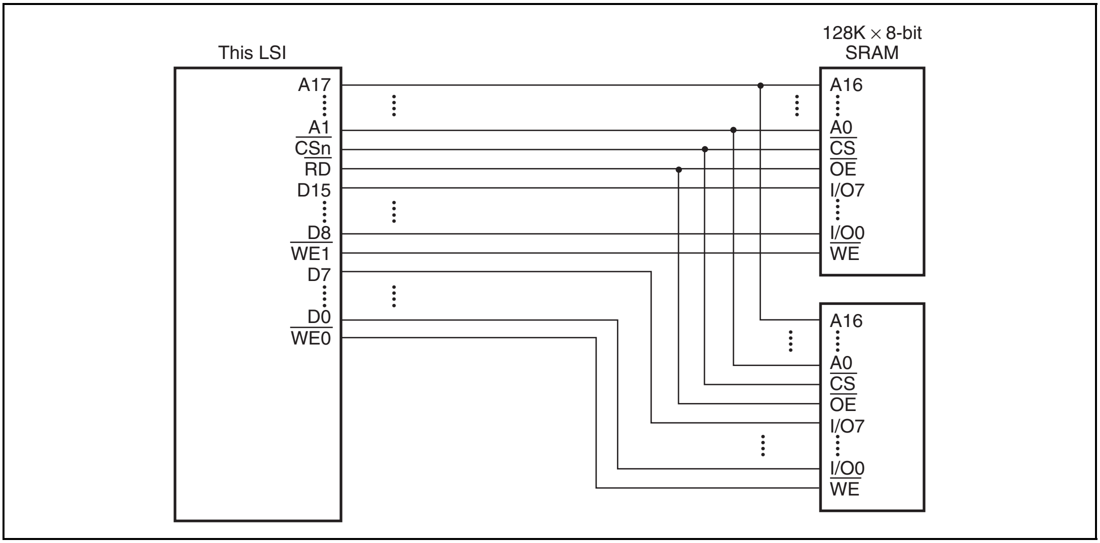


Figure 10.7 Example of 16-Bit Data-Width SRAM Connection

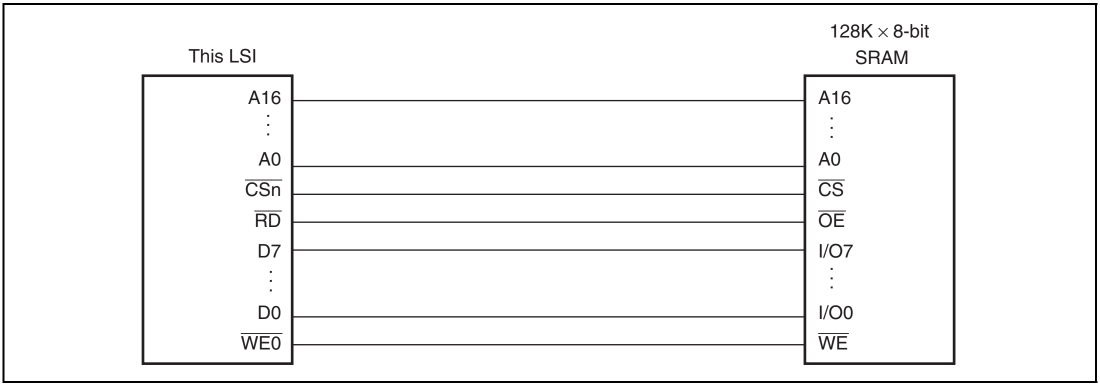


Figure 10.8 Example of 8-Bit Data-Width SRAM Connection

10.5.3 Access Wait Control

Wait cycle insertion on a normal space access can be controlled by the settings of bits WR3 to WR0 in CSnWCR. It is possible to insert wait cycles independently in read access and in write access. The specified number of Tw cycles are inserted as wait cycles in access shown in figure 10.9.

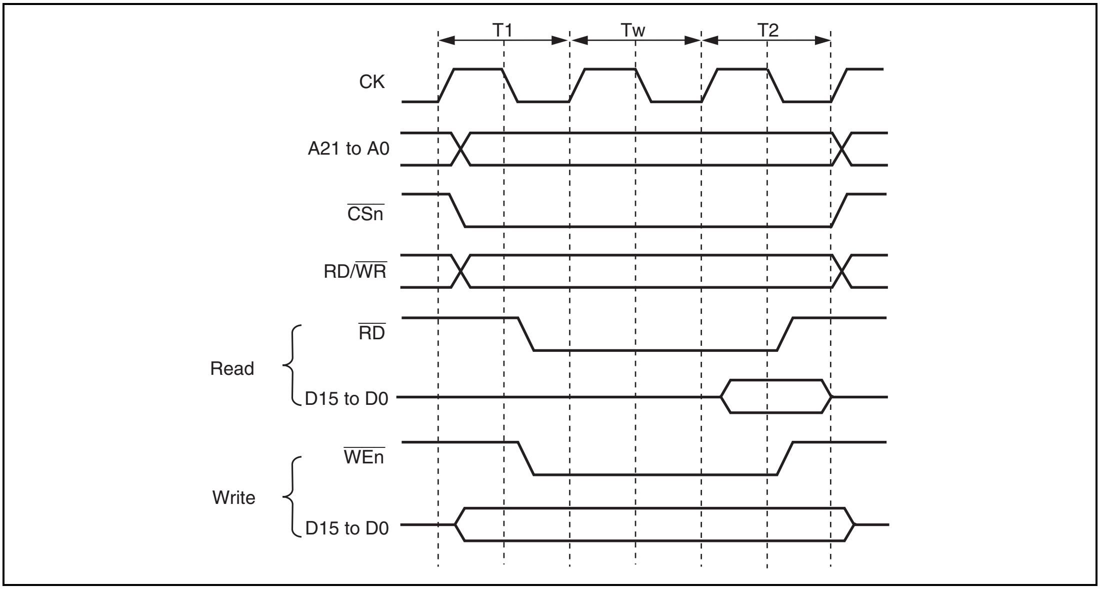


Figure 10.9 Wait Timing for External Space Access (Software Wait Only)

When the WM bit in CSnWCR is cleared to 0, the external wait input $\overline{\text{WAIT}}$ signal is also sampled. $\overline{\text{WAIT}}$ pin sampling is shown in figure 10.10. A 2-cycle wait is specified as a software wait. The $\overline{\text{WAIT}}$ signal is sampled on the falling edge of CK at the transition from the T1 or Tw cycle to the T2 cycle.

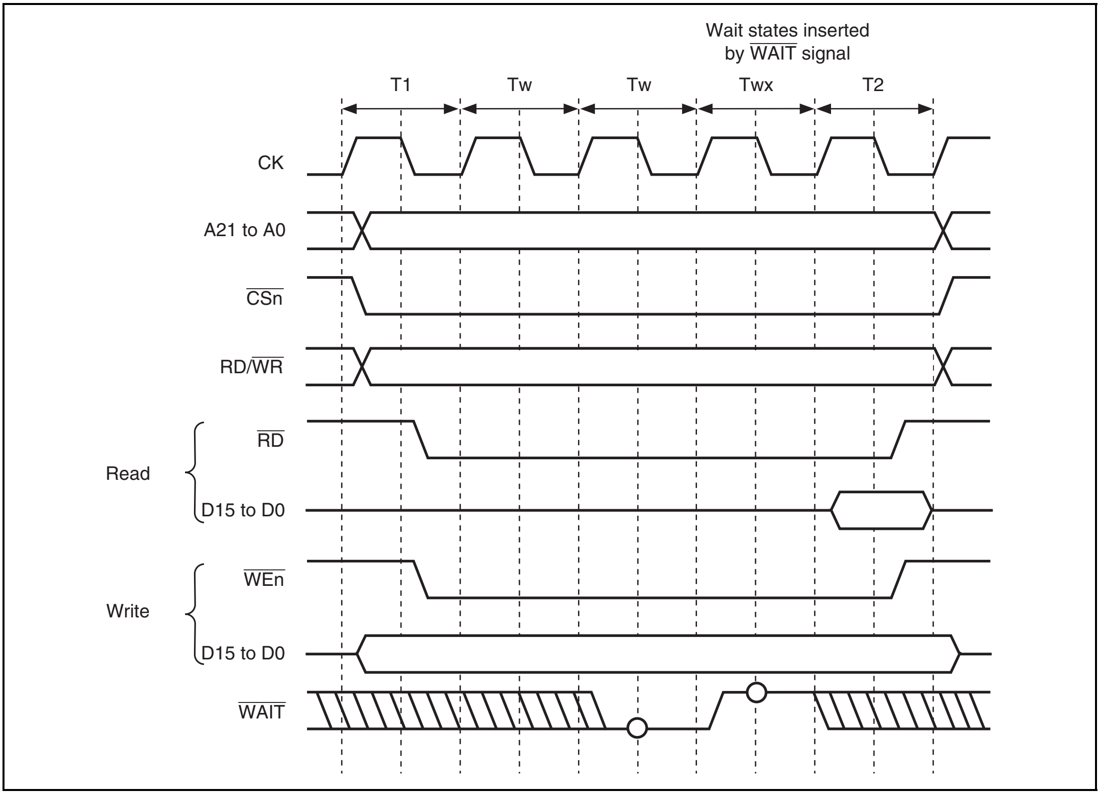


Figure 10.10 Wait Cycle Timing for External Space Access (Wait Cycle Insertion Using $\overline{\text{WAIT}}$ Signal)

10.5.4 $\overline{\text{CSn}}$ Assert Period Expansion

The number of cycles from $\overline{\text{CSn}}$ assertion to $\overline{\text{RD}}$, $\overline{\text{WEn}}$ assertion can be specified by setting bits SW1 and SW0 in CSnWCR. The number of cycles from $\overline{\text{RD}}$, $\overline{\text{WEn}}$ negation to $\overline{\text{CSn}}$ negation can be specified by setting bits HW1 and HW0. Therefore, a flexible interface to an external device can be obtained. Figure 10.11 shows an example. A Th cycle and a Tf cycle are added before and after an ordinary cycle, respectively. In these cycles, $\overline{\text{RD}}$ and $\overline{\text{WEn}}$ are not asserted, while other signals are asserted. The data output is prolonged to the Tf cycle, and this prolongation is useful for devices with slow writing operations.

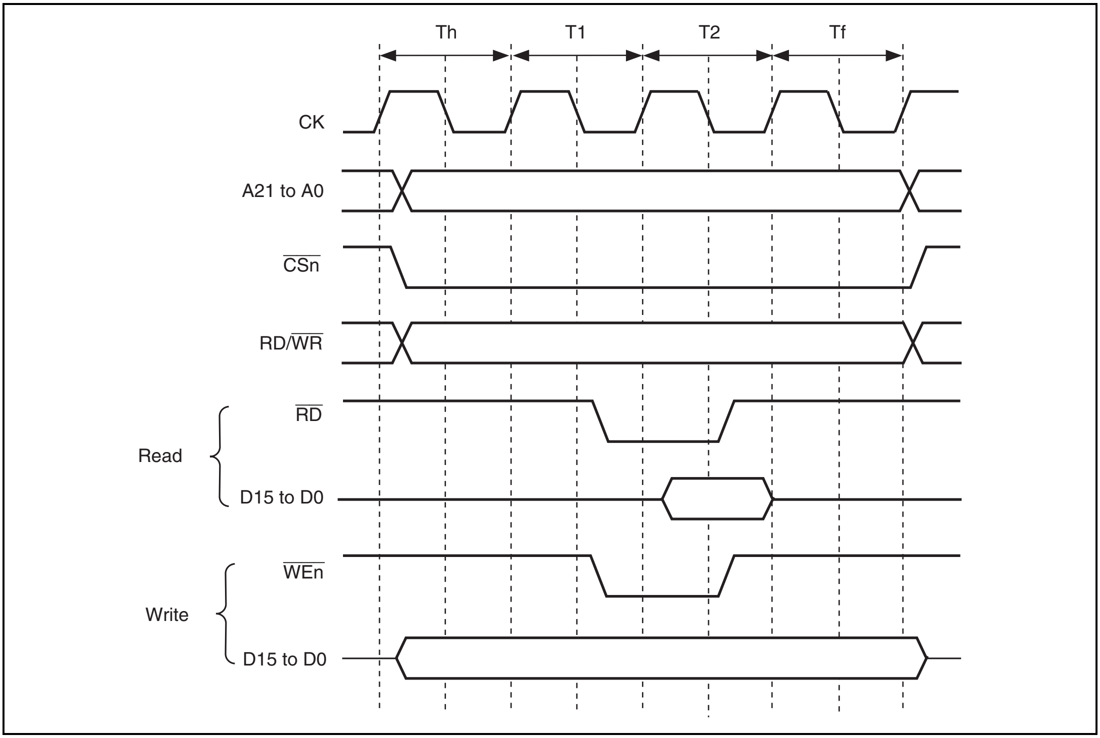


Figure 10.11 $\overline{\text{CSn}}$ Assert Period Expansion

Note: When two-times multiplication has been set for the peripheral clock, Tw must be equal to or more than one cycle and Tid must be equal to or more than two cycles.

10.5.5 SRAM Interface with Byte Selection

(1) Basic Timing for One-Time Multiplied Peripheral Clock

The SRAM interface with byte selection is for access to an SRAM which has a byte-selection pin ($\overline{\text{WEn}}$). This interface has 16-bit data pins and accesses SRAMs having upper and lower byte selection pins, such as UB and LB.

When the BAS bit in CSnWCR is cleared to 0 (initial value), the write access timing of the SRAM interface with byte selection is the same as that for the normal space interface. While in read access of a byte-selection SRAM interface, the byte-selection signal is output from the $\overline{\text{WEn}}$ pin, which is different from that for the normal space interface. The basic access timing is shown in figure 10.12. In write access, data is written to the memory according to the timing of the byte-selection pin ($\overline{\text{WEn}}$). For details, please refer to the Data Sheet for the corresponding memory.

If the BAS bit in CSnWCR is set to 1, the $\overline{\text{WEn}}$ pin and RD/$\overline{\text{WR}}$ pin timings change. Figure 10.13 shows the basic access timing. In write access, data is written to the memory according to the timing of the write enable pin (RD/$\overline{\text{WR}}$). The data hold timing from RD/$\overline{\text{WR}}$ negation to data write must be acquired by setting the HW1 and HW0 bits in CSnWCR. Figure 10.14 shows the access timing when a software wait is specified.

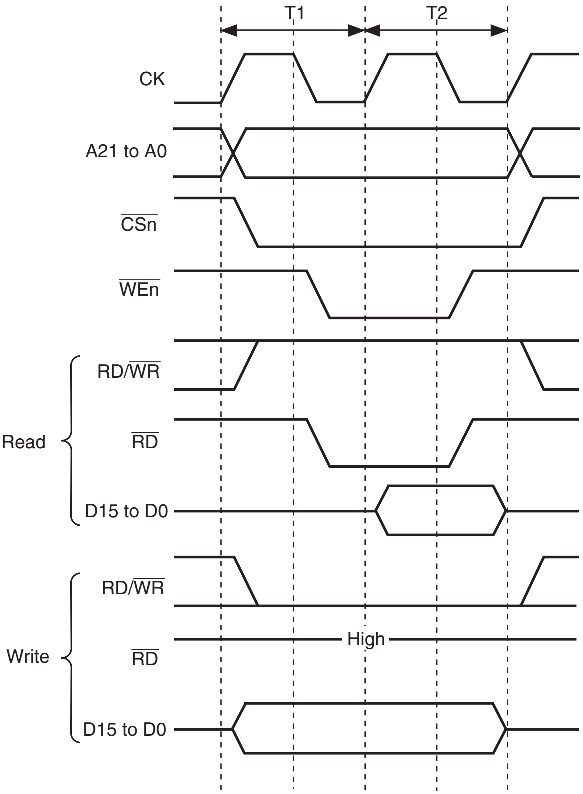


Figure 10.12 Basic Access Timing for SRAM with Byte Selection (BAS = 0)

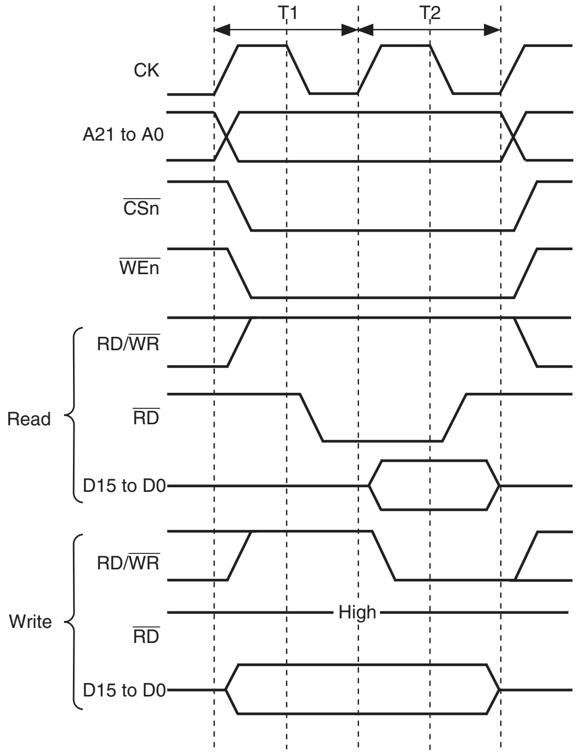


Figure 10.13 Basic Access Timing for SRAM with Byte Selection (BAS = 1)

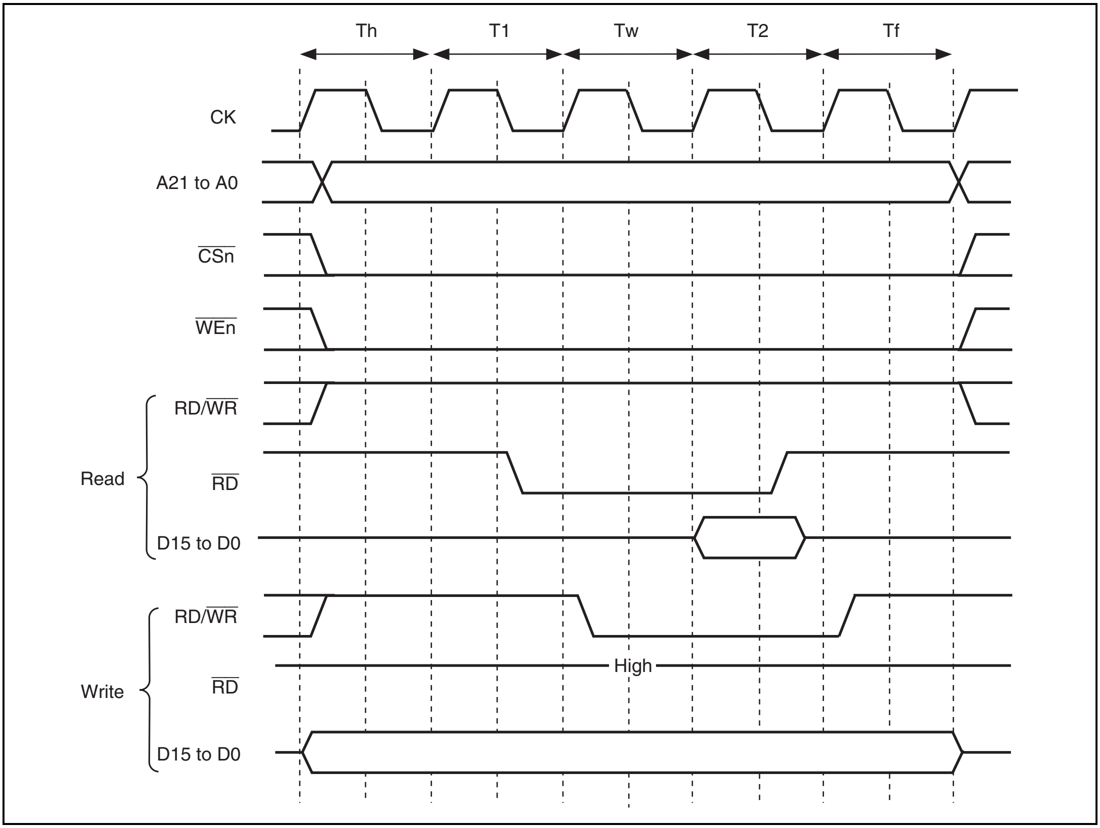


Figure 10.14 Wait Timing for SRAM with Byte Selection (BAS = 1) (SW[1:0] = 01, WR[3:0] = 0001, HW[1:0] = 01)

(2) Basic Timing for Two-Times Multiplied Peripheral Clock

When two-times multiplication has been set for the peripheral clock, Tw must be equal to or more than one cycle and Tid must be equal to or more than two cycles as with the normal space interface (for Tid, see figure 10.6). Figure 10.15 shows the basic access timing for the SRAM with byte selection (BAS = 0) on condition that two-times multiplication has been set for the peripheral clock. Figure 10.16 shows the basic access timing in the same situation as figure 10.15 except for BAS = 1.

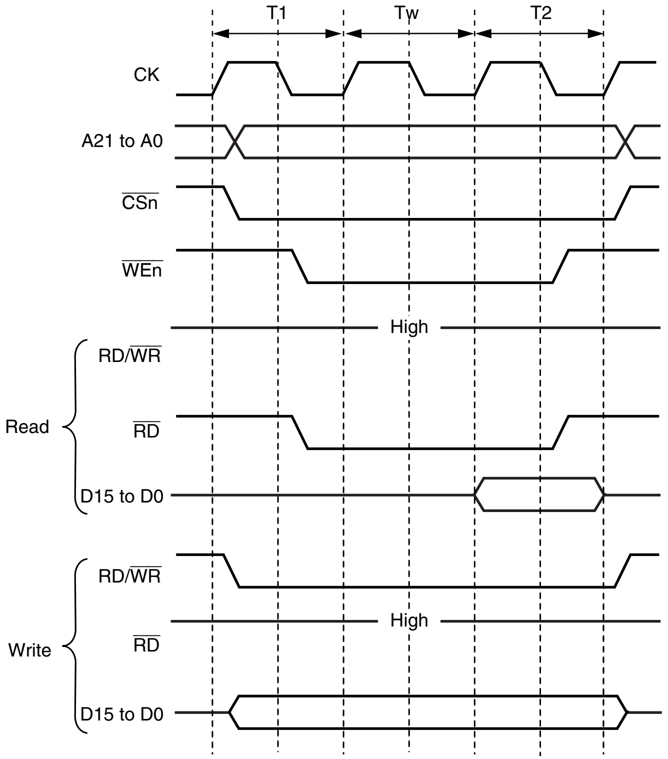


Figure 10.15 Basic Access Timing for SRAM with Byte Selection (BAS = 0) on Condition of Two-Times Multiplied Peripheral Clock

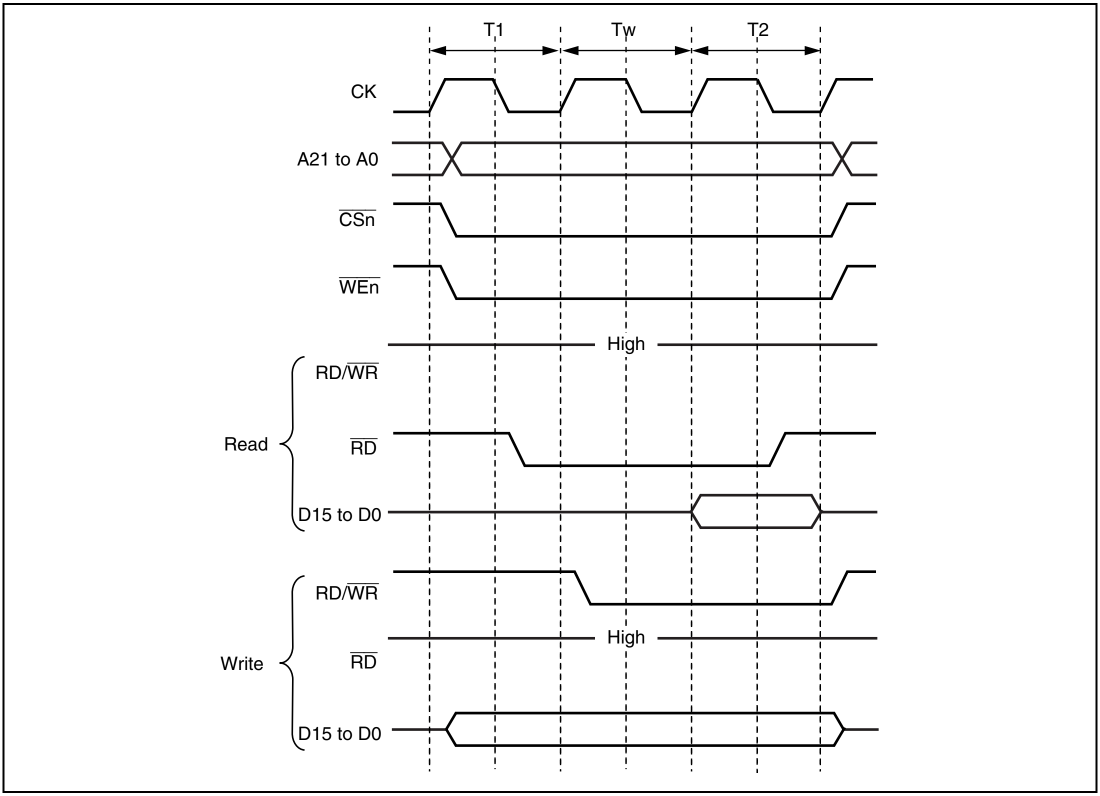


Figure 10.16 Basic Access Timing for SRAM with Byte Selection (BAS = 1) on Condition of Two-Times Multiplied Peripheral Clock

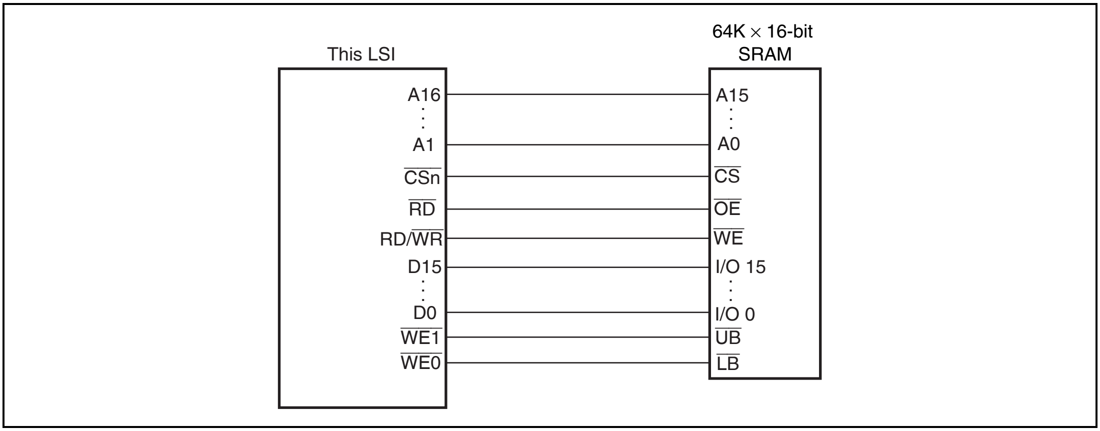


Figure 10.17 Example of Connection with 16-Bit Data-Width SRAM with Byte Selection

10.5.6 Wait between Access Cycles

As the operating frequency of LSIs becomes higher, the off-operation of the data buffer often collides with the next data access when the read operation from devices with slow access speed is completed. As a result of these collisions, the reliability of the device is low and malfunctions may occur. A function that avoids data collisions by inserting idle (wait) cycles between continuous access cycles has been newly added.

The number of wait cycles between access cycles can be set by bits IWW2 to IWW0, IWRWD2 to IWRWD0, IWRWS2 to IWRWS0, IWRRD2 to IWRRD0, and IWRRS2 to IWRRS 0 in CSnBCR. The conditions for setting the idle cycles between access cycles are shown below.

1. Continuous access cycles are write-read or write-write
2. Continuous access cycles are read-write for different spaces
3. Continuous access cycles are read-write for the same space
4. Continuous access cycles are read-read for different spaces
5. Continuous access cycles are read-read for the same space

For the specification of the number of idle cycles between access cycles described above, refer to the description of CSnBCR.

Besides the idle cycles between access cycles specified by the above-mentioned registers, idle cycles possibly are inserted to interface with the internal bus. The following gives detailed information about the number of idle cycles from $\overline{\text{CSn}}$ negation to $\overline{\text{CSn}}$ or $\overline{\text{CSm}}$ assertion.

There are six conditions that determine the number of idle cycles on the external bus as shown in table 10.8.

Table 10.8 Conditions for Determining Number of Idle Cycles

No.	Condition	Description	Range	Note
[1]	IW***[2:0] in CSnBCR	These bits specify the number of idle cycles for external address space access. The number of idle cycles can be specified independently for each combination of the previous and next cycles. For example, in the case where reading CS1 space followed by reading other CS space, the bits IWRRD[2:0] in CS1BCR should be set to B'100 to specify six or more idle cycles. This condition generates idle cycles after the access is completed.	0 to 12	Do not set 0 for the number of idle cycles between memory types which are not allowed to be accessed successively.
[2]	WM in CSnWCR	This bit enables or disables external $\overline{\text{WAIT}}$ pin. When this bit is set to B'0 (external $\overline{\text{WAIT}}$ enabled), one idle cycle is inserted to check the external $\overline{\text{WAIT}}$ pin input after the access is completed. When this bit is set to B'1 (disabled), no idle cycle is generated.	0 or 1	
[3]	Read data transfer cycle	One idle cycle is inserted after a read access is completed. This idle cycle is not generated for the first or middle cycles in divided access cycles. This is neither generated when the bits HW[1:0] in CSnWCR are set to other value than B'00.	0 or 1	

No.	Condition	Description	Range	Note
[4]	Internal bus idle cycles, etc.	External bus access requests from the CPU or DMAC and their results are passed through the internal bus. The external bus enters idle state during internal bus idle cycles or while a bus other than the external bus is being accessed. This condition is not effective for divided access cycles, which are generated by the BSC when the access size is larger than the external data bus width.	0 or larger	For CPU read/write → write access cycles, the minimum number of idle cycles is 1. For CPU read/write → read access cycles, the minimum number of idle cycles is 0. For DMAC read → write access cycles, the minimum number of idle cycles is 0.
[5]	Write data wait cycles	During write access, a write cycle is executed on the external bus only after the write data becomes ready. This write data wait period generates idle cycles before the write cycle. Note that when the previous cycle is a write cycle and the internal bus idle cycles are shorter than the previous write cycle, write data can be prepared in parallel with the previous write cycle and therefore, no idle cycle is generated (write buffer effect).	0 or 1	For write → write or write → read access cycles, successive access cycles without idle cycles are frequently available due to the write buffer effect described in the left column. If successive access cycles without idle cycles are not allowed, specify the minimum number of idle cycles between access cycles through CSnBCR.
[6]	Idle cycles between different memory types	To ensure the minimum pulse width on the signal-multiplexed pins, idle cycles may be inserted before access after memory types are switched.	0 or 1	One idle cycle is generated between SRAM access or SRAM with byte selection access while BAS = 0 and SRAM with byte selection access while BAS = 1.

In the above conditions, a total of four conditions, that is, condition [1], [2], a set of conditions [3] to [5] (these are generated successively, and therefore the sum of them should be taken as one set of idle cycles), and condition [6] are generated at the same time. The maximum number of idle cycles among these four conditions become the number of idle cycles on the external bus. To ensure the minimum idle cycles, be sure to make register settings for condition [1].

10.5.7 Others

(1) Reset

The bus state controller (BSC) can be initialized completely only at power-on reset. At power-on reset, all signals are immediately negated and data output buffers are turned off regardless of the bus cycle state. All control registers are initialized.

(2) Write Buffer Operation

Since the bus state controller (BSC) incorporates a one-stage write buffer, the BSC can execute an access via the internal bus before the previous external bus cycle is completed in a write cycle. If the on-chip module is read or written after the external low-speed memory is written, the on-chip module can be accessed before the completion of the external low-speed memory write cycle.

In read cycles, the CPU is placed in the wait state until read operation has been completed. To continue the process after the data write to the device has been completed, perform a dummy read to the same address to check for completion of the write before the next process to be executed.

The write buffer of the BSC functions in the same way for an access by a bus master other than the CPU such as the DMAC. Accordingly, to perform DMA transfer to external address spaces, the next read cycle is initiated before the previous write cycle is completed. Note, however, that if both the DMA source and destination addresses exist in external address space, the next write cycle will not be initiated until the previous write cycle is completed.

(3) On-Chip Peripheral Module Access

To access an on-chip module register, two or more peripheral module clock (Pϕ) cycles are required. Care must be taken in system design.

10.6 SRAM Access Timing

10.6.1 Standard SRAM

Figure 10.18 shows the SRAM access timing in the CS0 area which corresponds to the basic SRAM access timing shown in figure 10.2, and also shows the timing with wait cycles inserted. In the figure, "SW" denotes a delay cycle specified by the SW1 and SW0 bits in the CSnWCR register, "WW" shows a delay cycle specified by the WW2 to WW0 bits in the CSnWCR register, "HW" indicates a delay cycle specified by the HW1 and HW0 bits in the CSnWCR register, and "WR" denotes a delay cycle specified by the WR3 to WR0 bits in the CSnWCR register.

10.6.2 SRAM with Byte Selection (BAS = 0)

Figure 10.19 shows the SRAM access timing in the CS0 area which corresponds to the basic access timing for SRAM with byte selection shown in figure 10.12, and also shows the timing with wait cycles inserted. In the figure, "SW", "WW", "HW", and "WR" indicate the same delay cycles as those specified in section 10.6.1, Standard SRAM.

10.6.3 SRAM with Byte Selection (BAS = 1)

Figure 10.20 shows the SRAM access timing in the CS0 area which corresponds to he basic access timing for SRAM with byte selection shown in figure 10.13, and also shows the timing with wait cycles inserted. In the figure, "SW", "WW", "HW", and "WR" indicate the same delay cycles as those specified in section 10.6.1, Standard SRAM.

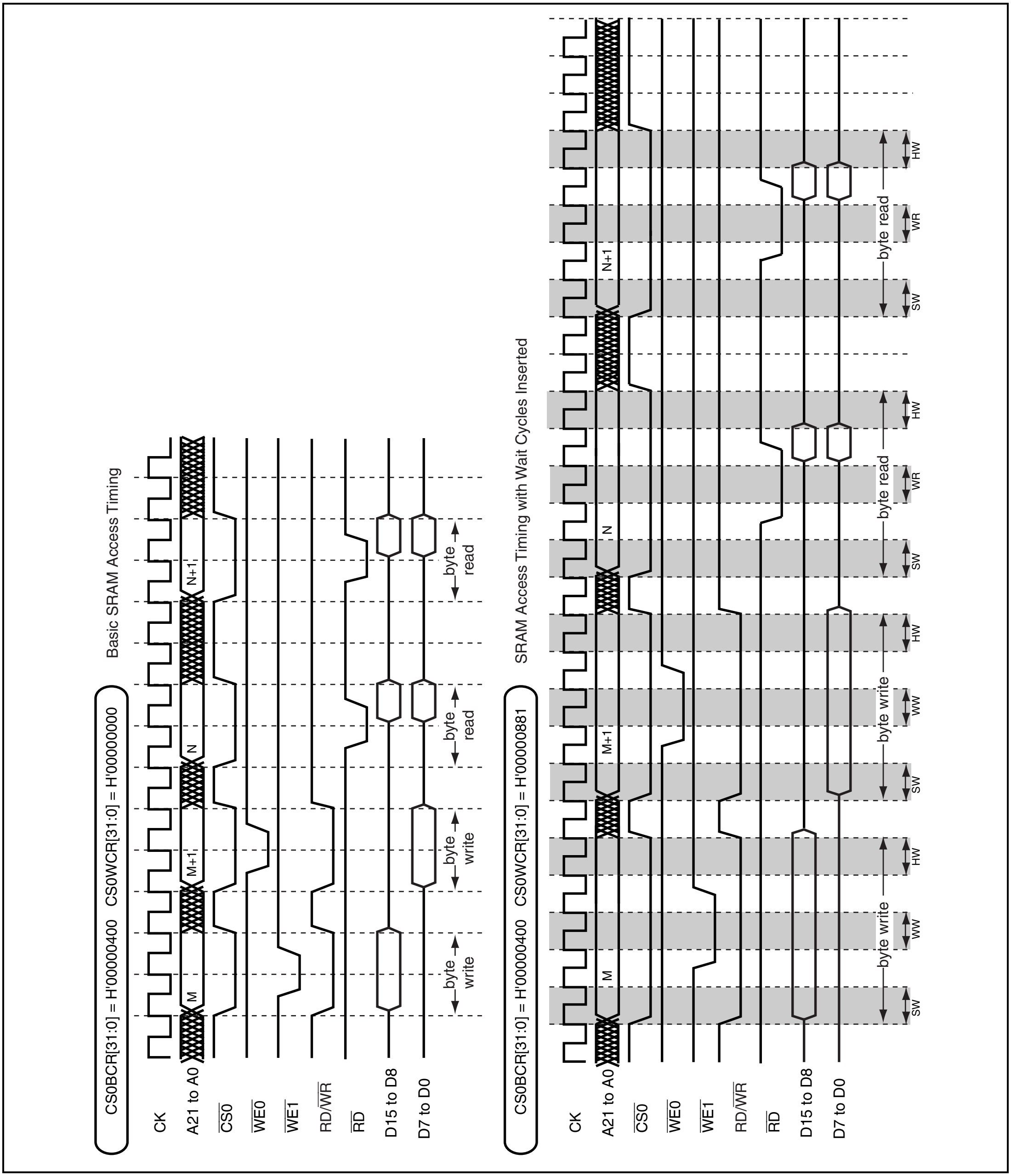


Figure 10.18 SRAM Access Timing (Basic Access Timing, with Wait Cycles)

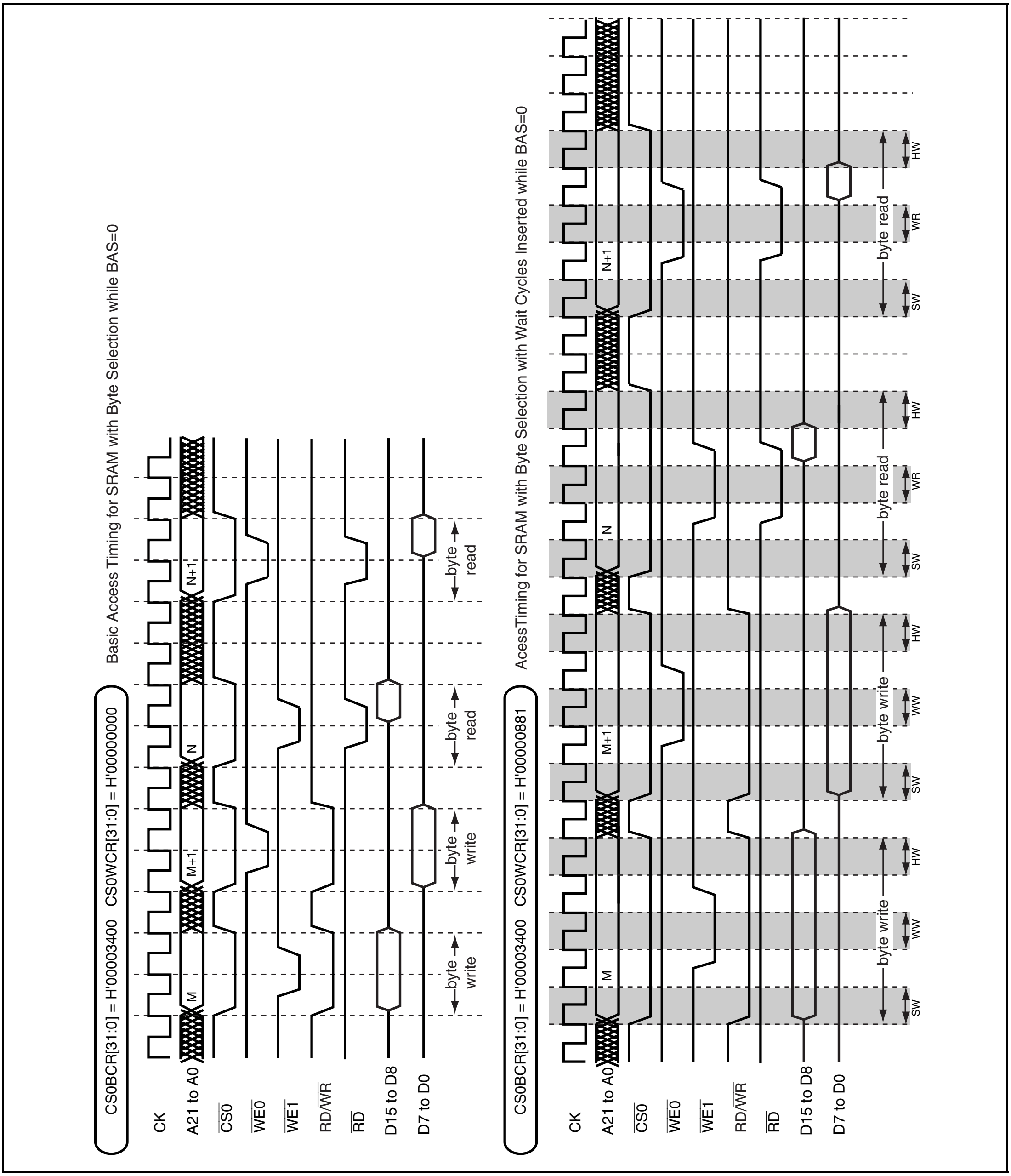

Basic Access Timing for SRAM with Byte Selection while BAS=0
CS0BCR[31:0] = H'00003400 CS0WCR[31:0] = H'00000000
CK
A21 to A0
CS0
WE0
WE1
RD/WR
RD
D15 to D8
D7 to D0
M
M+1
N
N+1
byte write
byte write
byte read
byte read
AcessTiming for SRAM with Byte Selection with Wait Cycles Inserted while BAS=0
CS0BCR[31:0] = H'00003400 CS0WCR[31:0] = H'00000881
CK
A21 to A0
CS0
WE0
WE1
RD/WR
RD
D15 to D8
D7 to D0
M
M+1
N
N+1
byte write
byte write
byte read
byte read
SW
WW
HW
SW
WW
HW
SW
WR
HW
SW
WR
HW

Figure 10.19 Access Timing for SRAM with Byte Selection while BAS=0 (Basic Access Timing, with Wait Cycles)

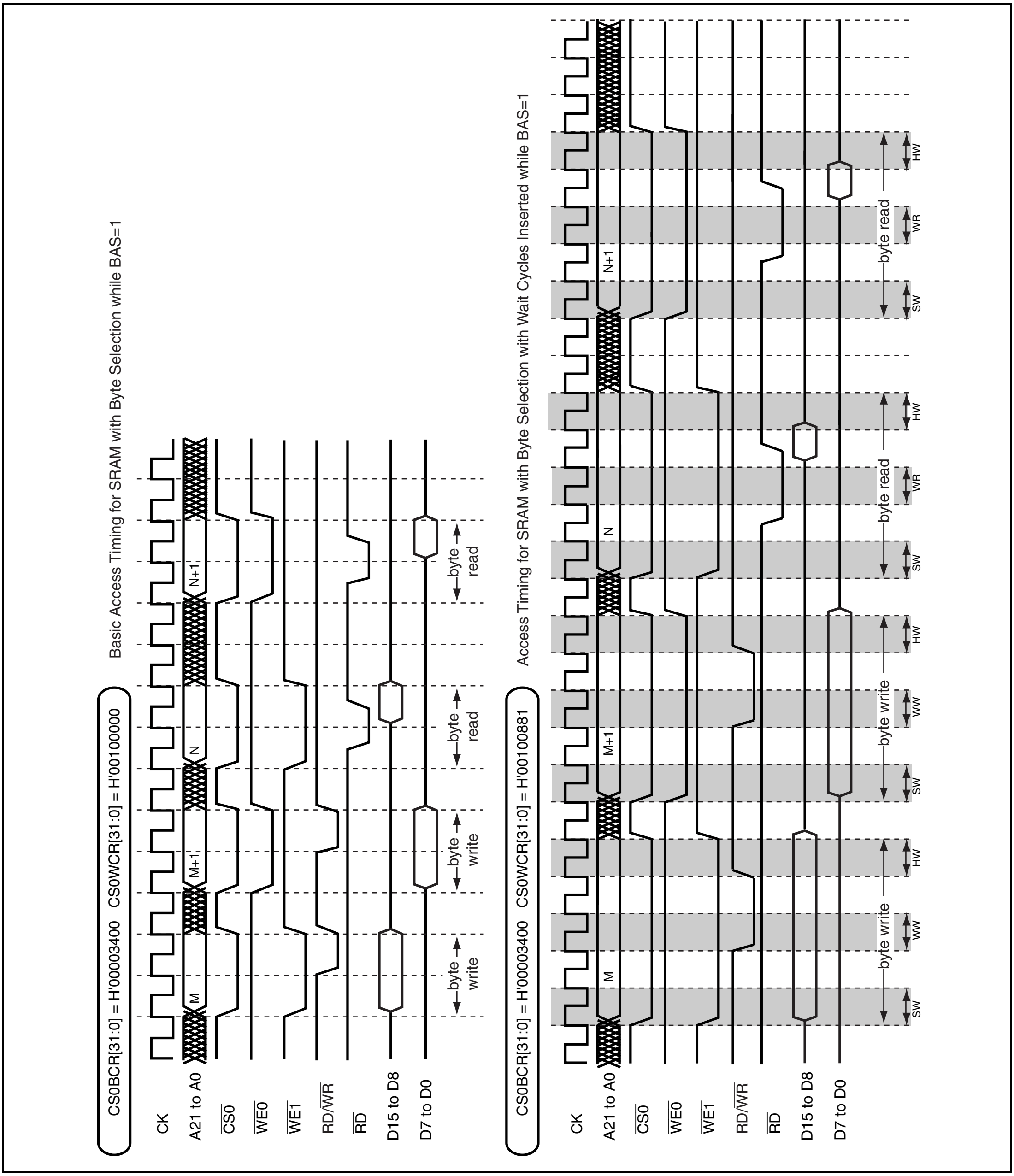

Basic Access Timing for SRAM with Byte Selection while BAS=1
CS0BCR[31:0] = H'00003400 CS0WCR[31:0] = H'00100000
CK
A21 to A0
CS0
WE0
WE1
RD/WR
RD
D15 to D8
D7 to D0
M
M+1
N
N+1
byte write
byte write
byte read
byte read
Access Timing for SRAM with Byte Selection with Wait Cycles Inserted while BAS=1
CS0BCR[31:0] = H'00003400 CS0WCR[31:0] = H'00100881
CK
A21 to A0
CS0
WE0
WE1
RD/WR
RD
D15 to D8
D7 to D0
M
M+1
N
N+1
byte write
byte write
byte read
byte read
SW
WW
HW
SW
WW
HW
SW
WR
HW
SW
WR
HW

Figure 10.20 Access Timing for SRAM with Byte Selection while BAS=1 (Basic Access Timing, with Wait Cycles)

Section 11 Direct Memory Access Controller (DMAC)

The DMAC can be used in place of the CPU to perform high-speed transfers between external memory, on-chip memory, external devices that are connected to the external address space, and on-chip peripheral modules.

11.1 Features

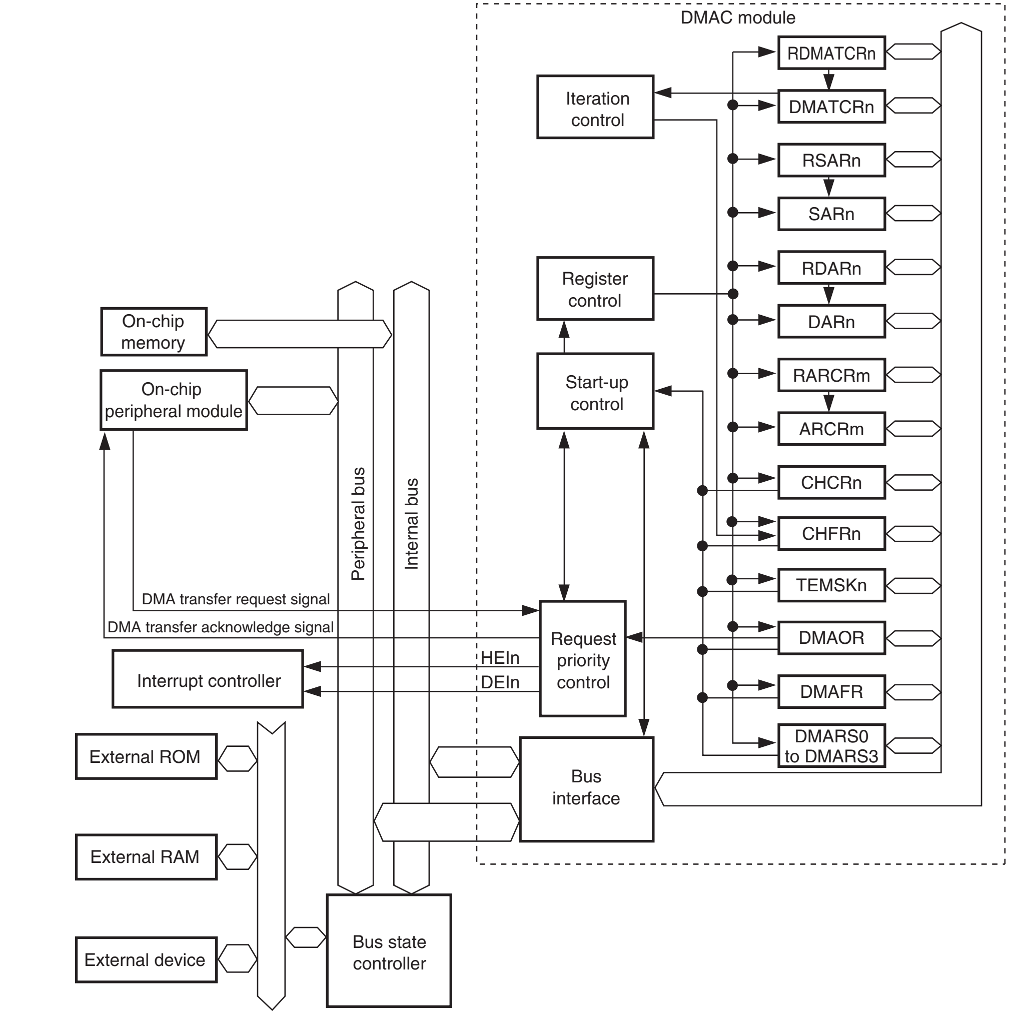
- Number of channels: Eight channels (channels 0 to 7)
- 4-Gbyte physical address space
- Data transfer unit: Selectable from byte, word (two bytes), longword (four bytes), and 16 bytes (longword × 4)
- Maximum transfer count: 16,777,216 transfers (24 bits)
- Address mode: Dual address mode
- Selectable transfer requests
  - On-chip peripheral module request
  - Auto request

  The following modules can issue on-chip peripheral module requests.
  - Four RCAN-TL1 sources, ten ADC sources, thirty-six ATU-III sources, and two CMT sources
- Selectable bus modes
  - Cycle stealing mode (normal mode and intermittent mode)
  - Burst mode
- Selectable channel priority levels
  - Two types of fixed modes
  - Round-robin mode
- Interrupt request: An interrupt request can be sent to the CPU on completion of half- or full-data transfer
- Register reloading functions: The reloading function can be enabled or disabled independently for each channel. Using this function, the values in the reload setting registers that can be set even during DMA transfer are transferred to the corresponding registers. The following two types of reloading functions are supported.
  - Reloading function 1: The DMA source address register, DMA destination address register, DMA transfer count register, and DMA address reload count register are reloaded when the transfer count register reaches 0. The address reload count registers on channels 4 to 7 are also reloaded when the reloading function 2 is enabled.

— Reloading function 2: The DMA source address register and/or DMA destination address register, and DMA address reload count register are reloaded when the address reload count register reaches 0. (Only supported in channels 4 to 7.)

- Continuous transfer when the reloading function 1 is enabled: Selectable between termination and continuation of transfer when the transfer counter reaches 0, on channels the reloading function 1 is enabled for.
- Integer to floating point conversion function: In the integer format to be converted, the data stored in 14 bits from the left edge of a 16-bit register is treated as unsigned data, and the decimal point is assumed to be located to the left of the MSB. After conversion, the floating point format is the single-precision floating-point format based on IEEE754, representing a 32-bit value.

Figure 11.1 shows a block diagram of the DMAC.



[Legend]

RDMATCR:	DMA reload transfer count register	CHFR:	DMA channel flag bit register
DMATCR:	DMA transfer count register	DMAOR:	DMA operation register
RSAR:	DMA reload source address register	DMAFR:	DMA operation flag bit register
SAR:	DMA source address register	TEMSK:	DMA TE flag mask setting register
RDAR:	DMA reload destination address register	DMARS0 to DMARS3:	DMA extension resource selectors 0 to 3
DAR:	DMA destination address register	HEIn:	DMA transfer half-end interrupt request to the CPU
RARCR:	DMA reload-address reload count register	DEIn:	DMA transfer end interrupt request to the CPU
ARCR:	DMA address reload count register	m:	4 to 7
CHCR:	DMA channel control register	n:	0 to 7

Figure 11.1 Block Diagram of DMAC

11.2 Register Descriptions

The DMAC has the registers listed in table 11.1. There are five (channels 0 to 3) or six (channels 4 to 7) control registers, three or four reload registers and one flag bit register for each channel, and one common operation register and one operation flag bit register are used by all channels. In addition, there is one extension resource selector per two channels. Notation for the registers in table 11.1 takes the form XXXN, where XXX indicates the register name and N indicates the channel number. For example, SAR0 denotes SAR for channel 0.

Table 11.1 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
0	DMA source address register 0	SAR0	R/W	H'00000000	H'FFFE1000	16, 32
	DMA destination address register 0	DAR0	R/W	H'00000000	H'FFFE1004	16, 32
	DMA transfer count register 0	DMATCR0	R/W	H'00000000	H'FFFE1008	16, 32
	DMA channel control register 0	CHCR0	R/W	H'00000000	H'FFFE100C	8, 16, 32
	DMA channel flag bit register 0	CHFR0	R/W*[1]	H'00	H'FFFE108C	8
	DMA TE flag mask setting register 0	TEMSK0	R/W*[2]	H'0000	H'FFFE108E	8, 16
	DMA reload source address register 0	RSAR0	R/W	H'00000000	H'FFFE1100	16, 32
	DMA reload destination address register 0	RDAR0	R/W	H'00000000	H'FFFE1104	16, 32
	DMA reload transfer count register 0	RDMATCR0	R/W	H'00000000	H'FFFE1108	16, 32

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
1	DMA source address register 1	SAR1	R/W	H'00000000	H'FFFE1010	16, 32
	DMA destination address register 1	DAR1	R/W	H'00000000	H'FFFE1014	16, 32
	DMA transfer count register 1	DMATCR1	R/W	H'00000000	H'FFFE1018	16, 32
	DMA channel control register 1	CHCR1	R/W	H'00000000	H'FFFE101C	8, 16, 32
	DMA channel flag bit register 1	CHFR1	R/W*[1]	H'00	H'FFFE109C	8
	DMA TE flag mask setting register 1	TEMSK1	R/W*[2]	H'0000	H'FFFE109E	8, 16
	DMA reload source address register 1	RSAR1	R/W	H'00000000	H'FFFE1110	16, 32
	DMA reload destination address register 1	RDAR1	R/W	H'00000000	H'FFFE1114	16, 32
	DMA reload transfer count register 1	RDMATCR1	R/W	H'00000000	H'FFFE1118	16, 32
2	DMA source address register 2	SAR2	R/W	H'00000000	H'FFFE1020	16, 32
	DMA destination address register 2	DAR2	R/W	H'00000000	H'FFFE1024	16, 32
	DMA transfer count register 2	DMATCR2	R/W	H'00000000	H'FFFE1028	16, 32
	DMA channel control register 2	CHCR2	R/W	H'00000000	H'FFFE102C	8, 16, 32
	DMA channel flag bit register 2	CHFR2	R/W*[1]	H'00	H'FFFE10AC	8
	DMA TE flag mask setting register 2	TEMSK2	R/W*[2]	H'0000	H'FFFE10AE	8, 16
	DMA reload source address register 2	RSAR2	R/W	H'00000000	H'FFFE1120	16, 32
	DMA reload destination address register 2	RDAR2	R/W	H'00000000	H'FFFE1124	16, 32
	DMA reload transfer count register 2	RDMATCR2	R/W	H'00000000	H'FFFE1128	16, 32

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
3	DMA source address register 3	SAR3	R/W	H'00000000	H'FFFE1030	16, 32
	DMA destination address register 3	DAR3	R/W	H'00000000	H'FFFE1034	16, 32
	DMA transfer count register 3	DMATCR3	R/W	H'00000000	H'FFFE1038	16, 32
	DMA channel control register 3	CHCR3	R/W	H'00000000	H'FFFE103C	8, 16, 32
	DMA channel flag bit register 3	CHFR3	R/W*[1]	H'00	H'FFFE10BC	8
	DMA TE flag mask setting register 3	TEMSK3	R/W*[2]	H'0000	H'FFFE10BE	8, 16
	DMA reload source address register 3	RSAR3	R/W	H'00000000	H'FFFE1130	16, 32
	DMA reload destination address register 3	RDAR3	R/W	H'00000000	H'FFFE1134	16, 32
	DMA reload transfer count register 3	RDMATCR3	R/W	H'00000000	H'FFFE1138	16, 32
4	DMA source address register 4	SAR4	R/W	H'00000000	H'FFFE1040	16, 32
	DMA destination address register 4	DAR4	R/W	H'00000000	H'FFFE1044	16, 32
	DMA transfer count register 4	DMATCR4	R/W	H'00000000	H'FFFE1048	16, 32
	DMA channel control register 4	CHCR4	R/W	H'00000000	H'FFFE104C	8, 16, 32
	DMA channel flag bit register 4	CHFR4	R/W*[1]	H'00	H'FFFE10CC	8
	DMA TE flag mask setting register 4	TEMSK4	R/W*[2]	H'0000	H'FFFE10CE	8, 16
	DMA reload source address register 4	RSAR4	R/W	H'00000000	H'FFFE1140	16, 32
	DMA reload destination address register 4	RDAR4	R/W	H'00000000	H'FFFE1144	16, 32
	DMA reload transfer count register 4	RDMATCR4	R/W	H'00000000	H'FFFE1148	16, 32

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
4	DMA address reload count register 4	ARCR4	R/W	H'0000	H'FFFE114C	16, 32
	DMA reload-address reload count register 4	RARCR4	R/W	H'0000	H'FFFE114E	16
5	DMA source address register 5	SAR5	R/W	H'00000000	H'FFFE1050	16, 32
	DMA destination address register 5	DAR5	R/W	H'00000000	H'FFFE1054	16, 32
	DMA transfer count register 5	DMATCR5	R/W	H'00000000	H'FFFE1058	16, 32
	DMA channel control register 5	CHCR5	R/W	H'00000000	H'FFFE105C	8, 16, 32
	DMA channel flag bit register 5	CHFR5	R/W*[1]	H'00	H'FFFE10DC	8
	DMA TE flag mask setting register 5	TEMSK5	R/W*[2]	H'0000	H'FFFE10DE	8, 16
	DMA reload source address register 5	RSAR5	R/W	H'00000000	H'FFFE1150	16, 32
	DMA reload destination address register 5	RDAR5	R/W	H'00000000	H'FFFE1154	16, 32
	DMA reload transfer count register 5	RDMATCR5	R/W	H'00000000	H'FFFE1158	16, 32
	DMA address reload count register 5	ARCR5	R/W	H'0000	H'FFFE115C	16, 32
	DMA reload-address reload count register 5	RARCR5	R/W	H'0000	H'FFFE115E	16
6	DMA source address register 6	SAR6	R/W	H'00000000	H'FFFE1060	16, 32
	DMA destination address register 6	DAR6	R/W	H'00000000	H'FFFE1064	16, 32
	DMA transfer count register 6	DMATCR6	R/W	H'00000000	H'FFFE1068	16, 32
	DMA channel control register 6	CHCR6	R/W	H'00000000	H'FFFE106C	8, 16, 32
	DMA channel flag bit register 6	CHFR6	R/W*[1]	H'00	H'FFFE10EC	8

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
6	DMA TE flag mask setting register 6	TEMSK6	R/W*[2]	H'0000	H'FFFE10EE	8, 16
	DMA reload source address register 6	RSAR6	R/W	H'00000000	H'FFFE1160	16, 32
	DMA reload destination address register 6	RDAR6	R/W	H'00000000	H'FFFE1164	16, 32
	DMA reload transfer count register 6	RDMATCR6	R/W	H'00000000	H'FFFE1168	16, 32
	DMA address reload count register 6	ARCR6	R/W	H'0000	H'FFFE116C	16, 32
	DMA reload-address reload count register 6	RARCR6	R/W	H'0000	H'FFFE116E	16
7	DMA source address register 7	SAR7	R/W	H'00000000	H'FFFE1070	16, 32
	DMA destination address register 7	DAR7	R/W	H'00000000	H'FFFE1074	16, 32
	DMA transfer count register 7	DMATCR7	R/W	H'00000000	H'FFFE1078	16, 32
	DMA channel control register 7	CHCR7	R/W	H'00000000	H'FFFE107C	8, 16, 32
	DMA channel flag bit register 7	CHFR7	R/W*[1]	H'00	H'FFFE10FC	8
	DMA TE flag mask setting register 7	TEMSK7	R/W*[2]	H'0000	H'FFFE10FE	8, 16
	DMA reload source address register 7	RSAR7	R/W	H'00000000	H'FFFE1170	16, 32
	DMA reload destination address register_7	RDAR7	R/W	H'00000000	H'FFFE1174	16, 32
	DMA reload transfer count register 7	RDMATCR7	R/W	H'00000000	H'FFFE1178	16, 32
	DMA address reload count register 7	ARCR7	R/W	H'0000	H'FFFE117C	16, 32
	DMA reload-address reload count register 7	RARCR7	R/W	H'0000	H'FFFE117E	16

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Common	DMA operation register	DMAOR	R/W	H'0000	H'FFFE1200	8, 16
	DMA operation flag bit register	DMAFR	R/W*[3]	H'00	H'FFFE1204	8
0 and 1	DMA extension resource selector 0	DMARS0	R/W	H'0000	H'FFFE1300	8, 16
2 and 3	DMA extension resource selector 1	DMARS1	R/W	H'0000	H'FFFE1304	8, 16
4 and 5	DMA extension resource selector 2	DMARS2	R/W	H'0000	H'FFFE1308	8, 16
6 and 7	DMA extension resource selector 3	DMARS3	R/W	H'0000	H'FFFE130C	8, 16

Notes: 1. Only writing a 0 to the HE and TE bits in CHFRn is allowed after reading a 1, in order to clear the flags.

2. TEMSKn includes a write key code. Write access is made only in word and allowed only if the upper byte, i.e. key code, matches H'5B.

3. Only writing a 0 to the AE and NMIF bits in DMAFR is allowed after reading a 1, in order to clear the flags.

11.2.1 DMA Source Address Registers 0 to 7 (SAR0 to SAR7)

SAR is a 32-bit readable/writable register that specifies the source address of a DMA transfer. During a DMA transfer, these registers indicate the next source address.

To transfer data in words (two bytes), in longwords (four bytes), or in 16-byte units, specify the address with 2-byte, 4-byte, or 16-byte address boundary.

SAR is initialized to H'00000000 by a power-on reset or a transition to the hardware standby mode and retains its value in module standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

11.2.2 DMA Destination Address Registers 0 to 7 (DAR0 to DAR7)

DAR is a 32-bit readable/writable register that specifies the destination address of a DMA transfer. During a DMA transfer, these registers indicate the next destination address.

To transfer data in words (two bytes), in longwords (four bytes), or in 16-byte units, specify the address with 2-byte, 4-byte, or 16-byte address boundary.

DAR is initialized to H'00000000 by a power-on reset or a transition to the hardware standby mode and retains its value in module standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

11.2.3 DMA Transfer Count Registers 0 to 7 (DMATCR0 to DMATCR7)

DMATCR is a 32-bit readable/writable register that specifies the number of DMA transfers. The transfer count is 1 when the setting is H'00000001, 16,777,215 when H'00FFFFFF is set, and 16,777,216 (the maximum) when H'00000000 is set. During a DMA transfer, these registers indicate the remaining transfer count.

The upper eight bits of DMATCR are always read as 0, and the write value should always be 0. To transfer data in 16-byte units, one 16-byte transfer (128 bits) counts one.

DMATCR is initialized to H'00000000 by a power-on reset or a transition to the hardware standby mode and retains its value in module standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-								
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

11.2.4 DMA Channel Control Registers 0 to 7 (CHCR0 to CHCR7)

CHCR is a 32-bit readable/writable register that controls the DMA transfer mode.

The TC[0], RLD2[1:0], and IFT bits can be read and written to in channels 4 to 7, but they are reserved in channels 0 to 3.

CHCR is initialized to H'00000000 by a power-on reset or a transition to the hardware standby mode and retains its value in module standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TC[1:0]		-	RLD1	RLD2[1:0]		-	IFT	-	-	-	-	-	HIE	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R	R	R	R	R	R/W	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DM[1:0]		SM[1:0]		RS[3:0]				-	-	TB	TS[1:0]		IE	-	DE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R	R/W

Bit	Bit Name	Initial Value	R/W	Descriptions
31, 30	TC[1:0]	00	R/W	Transfer Count Mode Specifies whether to transmit data once, for the count specified in DMATCR, or for the count specified in ARCR by one transfer request. The TC[0] bit is valid only in CHCR4 to CHCR7 and reserved in CHCR0 to CHCR3. The bit in CHCR0 to CHCR3 is always read as 0 and the write value should always be 0. When these bits are set to a value other than B'01, the TB bit must not be set to 1 (burst mode). When these bits are set to B'01, RLD2[1] and RLD2[0] bits should be a value other than B'00 to enable the reloading function 2. When these bits are set to B'00, operation is not guaranteed. 00: Transmits data once by one transfer request 01: Transmits data for the count specified in ARCR by one transfer request 10: Transmits data for the count specified in DMATCR by one transfer request 11: Setting prohibited
29	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
28	RLD1	0	R/W	Reloading Function 1 Enable or Disable Specifies whether the reloading function 1 is to be enabled or disabled. When this function is enabled, SAR, DAR, DMATCR or ARCR is reloaded when DMATCR is changed to 0. Note that ARCR is reloaded on channels 4-7 only when the reloading function 2 is also enabled. 0: Disables the reloading function 1 1: Enables the reloading functioning 1

Bit	Bit Name	Initial Value	R/W	Descriptions
27, 26	RLD2[1:0]	00	R/W	Reloading Function 2 Enable or Disable Specifies whether the reloading function 2 is to be enabled or disabled. When this function is enabled, SAR, DAR, or ARCR is reloaded when ARCR is changed to 0. These bits are valid only in CHCR4 to CHCR7; in CHCR0 to CHCR3, these bits are reserved. These bits are always read as 0 and the write value should always be 0. 00: Disables reloading function 2 01: Enables reloading function 2. DAR and ARCR are reloaded. 10: Enables reloading function 2. SAR and ARCR are reloaded. 11: Enables the reloading function 2. SAR, DAR, and ARCR are reloaded.
25	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Descriptions
24	IFT	0	R/W	Integer to Floating-Point Conversion Function Enable Specifies whether the function converting an integer to the floating-point format is enabled or disabled. The integer format in which the 14-bit data left-aligned in a 16-bit register is treated as unsigned data and the decimal point is assumed to be located to the left of the MSB can be treated. After conversion, the integer is represented by a 32-bit single-precision floating-point format based on IEEE 754 (the exponent has a bias of H'7F and the mantissa is represented by a form of 1.xxx, where the first digit 1 is an implicit leading bit). This bit is valid only in CHCR4 to CHCR7; in CHCR0 to CHCR3, this bit is reserved. This bit is always read as 0 and the write value should always be 0. If the integer to floating-point conversion function is enabled, the unit of data transfer should be specified as a word (TS in CHCR = B'01). In such a case, a 16-bit value is read as the source of transfer, an integer to floating-point conversion is performed, and the resulting 32-bit value is written. Consequently, the source register is updated by word access, and the destination address is updated by longword access. In addition, if the destination address is not a longword-aligned address, an address error may result. Notice that when the integer to floating-point conversion is enabled, the conversion requires cycles. Therefore, the internal operation within the DMAC will be read (conversion source to inside the DMAC) → write (inside the DMAC to conversion source), if a conversion is not required, and read → Nop (conversion processing) → write, if a conversion is required. 0: Disables the integer to floating-point conversion function 1: Enables the integer to floating-point conversion function
23 to 19	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Descriptions
18	HIE	0	R/W	Half-End Interrupt Enable Specifies whether to issue an interrupt request to the CPU when the transfer count reaches half of the DMATCR value that was specified before transfer starts. When the HE bit in CHFR is set to 1, the DMAC requests an interrupt to the CPU when the HE bit becomes 1. 0: Disables an interrupt to be issued when DMATCR = (DMATCR set before transfer starts)/2 1: Enables an interrupt to be issued when DMATCR = (DMATCR set before transfer starts)/2
17, 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15, 14	DM[1:0]	00	R/W	Destination Address Mode These bits select whether the DMA destination address is incremented, decremented, or left fixed. 00: Fixed destination address (Setting prohibited in 16-byte transfer) 01: Destination address is incremented (+1 in 8-bit transfer, +2 in 16-bit transfer, +4 in 32-bit transfer, +16 in 16-byte transfer) 10: Destination address is decremented (–1 in 8-bit transfer, –2 in 16-bit transfer, –4 in 32-bit transfer, setting prohibited in 16-byte transfer) 11: Setting prohibited
13, 12	SM[1:0]	00	R/W	Source Address Mode These bits select whether the DMA source address is incremented, decremented, or left fixed. 00: Fixed source address (Setting prohibited in 16-byte-unit transfer) 01: Source address is incremented (+1 in byte-unit transfer, +2 in word-unit transfer, +4 in longword-unit transfer, +16 in 16-byte-unit transfer) 10: Source address is decremented (–1 in byte-unit transfer, –2 in word-unit transfer, –4 in longword-unit transfer, setting prohibited in 16-byte-unit transfer) 11: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Descriptions
11 to 8	RS[3:0]	0000	R/W	Resource Select These bits specify which transfer requests will be sent to the DMAC. The changing of transfer request source should be done in the state when DMA enable bit (DE) is cleared to 0. 0000: Initial value (when the resource is not selected) 0001: Setting prohibited 0010: Setting prohibited 0011: Setting prohibited 0100: Auto request 0101: Setting prohibited 0110: Setting prohibited 0111: Setting prohibited 1000: DMA extension resource selector 1001: RCAN-TL1 channel A 1010: RCAN-TL1 channel B 1011: RCAN-TL1 channel C 1100: ADC_A (A/D conversion end) 1101: ADC_B (A/D conversion end) 1110: RCAN-TL1 channel D 1111: Setting prohibited
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	TB	0	R/W	Transfer Bus Mode Specifies the bus mode when DMA transfers data. Note that the burst mode must not be selected when TC is other than B'10. 0: Cycle stealing mode 1: Burst mode
4, 3	TS[1:0]	00	R/W	Transfer Size These bits specify the size of data to be transferred. Select the size of data to be transferred when the source or destination is an on-chip peripheral module register of which transfer size is specified. 00: Byte units 01: Word (two bytes) units 10: Longword (four bytes) units 11: 16-byte units (four longword transfers)

Bit	Bit Name	Initial Value	R/W	Descriptions
2	IE	0	R/W	Interrupt Enable Specifies whether or not an interrupt request is generated to the CPU at the end of the DMA transfer. Setting this bit to 1 generates an interrupt request (DEI) to the CPU when the TE bit in CHFR is set to 1. 0: Disables an interrupt request 1: Enables an interrupt request
1	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
0	DE	0	R/W	DMA Enable Enables or disables the DMA transfer. In auto request mode, DMA transfer starts by setting the DE bit and DME bit in DMAOR to 1. In this case, all of the bits TE in CHFR, NMIF and AE in DMAFR must be 0. In a peripheral module request, DMA transfer starts if DMA transfer request is generated by peripheral modules after setting the bits DE and DME to 1. In this case, however, all of the bits TE in CHFR, NMIF and AE in DMAFR must be 0 as in the case of auto request mode. Clearing the DE bit to 0 can terminate the DMA transfer. 0: DMA transfer disabled 1: DMA transfer enabled

11.2.5 DMA Channel Flag Bit Registers 0 to 7 (CHFR0 to CHFR7)

CHFR is an 8-bit readable/writable register that shows the DMA transfer result.

CHFR is initialized to H'00 by a power-on reset or a transition to the hardware standby mode and retains its value in module standby mode.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	HE	-	-	-	TE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/(W)*	R	R	R	R/(W)*

Note: * To clear the flag, write 0 to the bit to be cleared, only when 1 has been read from the bit when these registers have been read. To the HE or TE bit not to be cleared, only 1 can be written, even if 0 is read from the bit. To the reserved bits, only 0 can be written. When the flag is read by the CPU, even though 0 has been read, the flag may be set to 1 and therefore 1 may be read internally. In this case, if 0 is written to the corresponding bit, the flag will be cleared even if 1 has not been read by the CPU. Since the specifications of the flag bits in these registers differ from the ones of the flags in other modules, the clearing operations of the flags by the CPU must be executed carefully.

Bit	Bit Name	Initial Value	R/W	Descriptions
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	HE	0	R/(W)*	Half-End Flag This bit is set to 1 when the transfer count reaches half of the DMATCR value specified before transfer starts. When DMA transfer ends because of an NMI interrupt, a DMA address error, or clearing of the DE bit or the DME bit in DMAOR before the transfer count reaches half of the initial DMATCR value, the HE bit is not set to 1. If DMA transfer ends due to an NMI interrupt, a DMA address error, or clearing of the DE bit or the DME bit in DMAOR after the HE bit is set to 1, the bit remains set to 1. To clear the HE bit, write 0 to it after HE = 1 is read. Note that the HE bit is never set even after it is cleared when the transfer count is equal to or more than the half of the initial DMATCR value. 0: DMATCR > (DMATCR set before transfer starts)/2 during DMA transfer or after DMA transfer is terminated 1: DMATCR ≤ (DMATCR set before transfer starts)/2 [Clearing condition] • Writing 0 after reading HE = 1.
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Descriptions
0	TE	0	R/(W)*	Transfer End Flag This bit is set to 1 when DMATCR becomes 0 and DMA transfer ends. The TE bit is not set to 1 in the following cases. • When DMA transfer ends due to an NMI interrupt or DMA address error before DMATCR becomes 0 • When DMA transfer is ended by clearing the DE bit and DME bit in DMA operation register (DMAOR) To clear the TE bit, write 0 after reading TE = 1. Even if the DE bit is set to 1 while this bit is set to 1, transfer is not enabled. 0: During the DMA transfer or DMA transfer has been terminated 1: DMA transfer ends by the specified count (DMATCR = 0) [Clearing condition] • Writing 0 after reading TE = 1

Note: * Only writing a 0 is allowed after reading a 1, in order to clear the flag.

11.2.6 DMA TE Flag Mask Setting Registers 0 to 7 (TEMSK0 to TEMSK7)

TEMSK is a 16-bit readable/writable register, which contains an 8-bit write key. If the reloading function 1 is enabled, the TE flag setting controls whether to terminate or continue DMA transfer. If the reloading function 1 is disabled, the resister setting is ignored and DMA transfer is terminated when the TE flag is set.

The register is initialized to H'0000 by a power-on reset or a transition to the hardware standby mode but retains its value in module standby mode.

Since TEMSK includes a write key, a write to this register must be performed in word. To rewrite the TEMASK bit value, write a value of H'5B to the TEMKEY in advance. A write in word when a value other than H'5B is written to the TEMKEY bit and a write in byte are all ignored.

A read out from TEMSK can be performed both in word and byte. However, the TEMKEY bits are always read as H'00 since no write is held in these bits.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TEMKEY[7:0]								-	-	-	-	-	-	-	TEM ASK
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R	R	R	R	R	R	R	R/W

Note: * Write data is not retained; these bits are always read as 0.

Bit	Bit Name	Initial Value	R/W	Descriptions
15 to 8	TEMKEY [7:0]	All 0	R/W*	TEMSK Write Key Code These bits act as the write key for TEMSK that enables/disables rewriting the TEMASK bit. Since no write value is held, these bits are always read as H'00. H'5B: Enables rewriting the TEMASK bit. Other than H'5B: Disables rewriting the TEMASK bit.
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Descriptions
0	TEMASK	0	R/W	TE Flag Mask Setting Selects whether or not to terminate DMA transfer when the TE bit is set to 1. By setting this bit to 1 while the reloading function 1 is enabled, DMA transfer is performed until its transfer request is canceled. This function can only be enabled when the reloading function 1 is enabled. When DMA transfer is continued even after the TE flag is set to 1 by setting the TEMASK bit to 1 while reloading function 1 is enabled, the operation of the HE and TE bits of the CHFR register is as follows: • If the HE bit is cleared in the middle of transfer, the bit is again set when the DMATCR value has become equal to or less than the half of its value set before transfer starts. • If the TE bit is cleared in the middle of transfer, the bit is again set when the DMATCR value has become 0 next time. • The information of that the set HE or TE bit was read for clearing is held until the bit is cleared. Therefore, the set HE or TE bit that was read can be cleared by simply writing a 0 to it even after the register values are changed by the reloading function 1. 0: Terminates DMA transfer when the TE flag is set to 1. 1: Continues DMA transfer even when the TE flag is set to 1. Note: When this function is enabled, care must be taken since the bus occupancy of DMA transfer becomes higher. In particular, when auto request is selected as the request type, do not select burst transfer as the bus mode, as correct operation is not guaranteed in this case.

11.2.7 DMA Reload Source Address Registers 0 to 7 (RSAR0 to RSAR7)

RSAR is a 32-bit readable/writable register that can be modified even during DMA transfer.

When the reloading function 1 is enabled (RLD1 in CHCR = 1), the RSAR value is written to the source address register (SAR) at the end of the current DMA transfer (DMATCR = 0). When the reloading function 2 is enabled (RLD2[1] bit in CHCR = 1), the RSAR value is written to the source address register (SAR) when the current address reload count register reaches 0 (ARCR = 0). A new value for the next DMA transfer can be preset in RSAR. When both of the reloading functions are disabled, RSAR is ignored.

To transfer data in words (two bytes), in longwords (four bytes), or in 16-byte units, specify the address with 2-byte, 4-byte, or 16-byte address boundary.

RSAR is initialized to H'00000000 by a power-on reset or a transition to the hardware standby mode and retains its value in module standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

11.2.8 DMA Reload Destination Address Registers 0 to 7 (RDAR0 to RDAR7)

RDAR is a 32-bit readable/writable register that can be modified even during DMA transfer.

When the reloading function 1 is enabled (RLD1 in CHCR = 1), the RDAR value is written to the destination address register (DAR) at the end of the current DMA transfer (DMATCR = 0). When the reloading function 2 is enabled (RLD2[0] bit in CHCR = 1), the RDAR value is written to the destination address register (DAR) when the current address reload count register reaches 0 (ARCR = 0). A new value for the next DMA transfer can be preset in RDAR. When both of the reloading functions are disabled, RDAR is ignored.

To transfer data in words (two bytes), in longwords (four bytes), or in 16-byte units, specify the address with 2-byte, 4-byte, or 16-byte address boundary.

RDAR is initialized to H'00000000 by a power-on reset or a transition to the hardware standby mode and retains its value in module standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

11.2.9 DMA Reload Transfer Count Registers 0 to 7 (RDMATCR0 to RDMATCR7)

RDMATCR is a 32-bit readable/writable register that can be modified even during DMA transfer.

When the reloading function 1 is enabled (RLD1 in CHCR = 1), the RDMATCR value is written to the transfer count register (DMATCR) at the end of the current DMA transfer (DMATCR = 0). A new value for the next DMA transfer can be preset in RDMATCR. When the reloading function 1 is disabled (RLD1 in CHCR = 0), RDMATCR is ignored.

The upper eight bits of RDMATCR are always read as 0, and the write value should always be 0.

As in DMATCR, the transfer count is 1 when the setting is H'00000001, 16,777,215 when H'00FFFFFF is set, and 16,777,216 (the maximum) when H'00000000 is set. To transfer data in 16-byte units, one 16-byte transfer (128 bits) counts one.

RDMATCR is initialized to H'00000000 by a power-on reset or a transition to the hardware standby mode and retains its value in module standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-								
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

11.2.10 DMA Address Reload Count Registers 4 to 7 (ARCR4 to ARCR7)

ARCR is a 16-bit readable/writable register.

When the reloading function 2 is enabled (CHCR.RLD2[1:0] = 1), the ARCR register and the source address register (SAR) and/or the destination address register (DAR) are reloaded once ARCR reaches 0. When the reloading function 2 is disabled (CHCR.RLD2[1:0] = 0), this register is ignored. Note that the ARCR register is reloaded only when the reloading function 2 is enabled (CHCR.RLD2[1:0] = 1).

The upper eight bits in ARCR are always read as 0 and the write value should always be 0.

As in DMATCR, the transfer count is 1 when the setting is H'0001, 255 when H'00FF is set, and 256 (the maximum) when H'0000 is set. To transfer data in 16-byte units, one 16-byte transfer (128 bits) counts one.

ARCR is initialized to H'0000 by a power-on reset or a transition to the hardware standby mode and retains its value in module standby mode.

When the reloading function 2 is enabled (RLD1 or RLD2[0] bit in CHCR = 1), ARCR and SAR and/or DAR are reloaded when ARCR reaches 0. When the reloading function 2 is disabled (RLD1 and RLD2[0] bits are 0), ARCR is ignored. ARCR is updated only when the reloading function 2 is enabled.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-								
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

11.2.11 DMA Reload-Address Reload Count Registers 4 to 7 (RARCR4 to RARCR7)

RARCR is a 16-bit readable/writable register that can be modified even during DMA transfer.

When the reloading function 2 is enabled (RLD1 or RLD2[0] bit in CHCR = 1), the RARCR value is written to the DMA address reload count register (ARCR) when the current reload count register reaches 0 (ARCR = 0). A new value for the next DMA transfer can be preset in RARCR. When the reloading function 2 is disabled (RLD1 and RLD2[0] bits in CHCR are 0), RARCR is ignored.

When both reloading functions are enabled, the contents of RARCR is written to the address reload count register (ARCR) on completion of the current DMA transfer (DMATCR = 0).

The upper eight bits of RARCR are always read as 0, and the write value should always be 0.

As in ARCR, the transfer count is 1 when the setting is H'0001, 255 when H'00FF is set, and 256 (the maximum) when H'0000 is set. To transfer data in 16-byte units, one 16-byte transfer (128 bits) counts one.

RARCR is initialized to H'0000 by a power-on reset or a transition to the hardware standby mode and retains its value in module standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-								
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

11.2.12 DMA Operation Register (DMAOR)

DMAOR is a 16-bit readable/writable register that specifies the priority level of channels at the DMA transfer.

DMAOR is initialized to H'0000 by a power-on reset or a transition to the hardware standby mode and retains its value in module standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	CMS[1:0]		-	-	PR[1:0]		-	-	-	-	-	-	-	DME
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13, 12	CMS[1:0]	00	R/W	Cycle Stealing Mode Select These bits select either normal mode or intermittent mode in cycle stealing mode. It is necessary that the bus modes of all channels be set to cycle stealing mode to make the intermittent mode valid. 00: Normal mode 01: Setting prohibited 10: Intermittent mode 16 One DMA transfer for every 16 cycles of Pϕ clock. 11: Intermittent mode 64 One DMA transfer for every 64 cycles of Pϕ clock.

Bit	Bit Name	Initial Value	R/W	Description
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	PR[1:0]	00	R/W	Priority Mode These bits select the priority level between channels when there are transfer requests for multiple channels simultaneously. 00: Fixed mode 1: CH0 > CH1 > CH2 > CH3 > CH4 > CH5 > CH6 > CH7 01: Fixed mode 2: CH0 > CH4 > CH1 > CH5 > CH2 > CH6 > CH3 > CH7 10: Setting prohibited 11: Round-robin mode (only supported in CH0 to CH3)
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	DME	0	R/W	DMA Master Enable Enables or disables DMA transfer on all channels. If the DME bit and DE bit in CHCR are set to 1, DMA transfer is enabled. However, transfer is enabled only when the TE bit in CHFR is set to 0 or the RLD1 bit in CHCR and TEMASK bit are all set to 1, of the corresponding channel for transfer. Also, the NMIF and AE bits in DMAFR must be all cleared to 0. Clearing the DME bit to 0 discontinues the DMA transfer on all channels. 0: DMA transfer is disabled on all channels 1: DMA transfer is enabled on all channels

If the priority mode bits are modified after a DMA transfer, the channel priority is initialized. If fixed mode 2 is specified, the channel priority is specified as CH0 > CH4 > CH1 > CH5 > CH2 > CH6 > CH3 > CH7. If fixed mode 1 is specified, the channel priority is specified as CH0 > CH1 > CH2 > CH3 > CH4 > CH5 > CH6 > CH7. If the round-robin mode is specified, the transfer end channel is reset.

Table 11.2 shows the priority change in each mode (modes 0 to 2) specified by the priority mode bits. In round-robin mode, the channel priority to accept the next transfer request may change in up to three ways according to the transfer end channel.

For example, when the transfer end channel is channel 1, the priority of the channel to accept the next transfer request is specified as CH2 > CH3 > CH0 >CH1 > CH4 > CH5 > CH6 > CH7. When the transfer end channel is any one of the channels 4 to 7, round-robin will not be applied and the priority level is not changed at the end of transfer in the channels 4 to 7.

Table 11.2 Combinations of Priority Mode Bits

	Transfer End	Priority Mode Bits		High ←	Priority Level at End of Transfer						→ Low
Mode	**CH No.**	**PR1**	**PR0**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Mode 0 (fixed mode 1)	Any channel	0	0	CH0	CH1	CH2	CH3	CH4	CH5	CH6	CH7
Mode 1 (fixed mode 2)	Any channel	0	1	CH0	CH4	CH1	CH5	CH2	CH6	CH3	CH7
Mode 2 (round-robin mode)	CH0	1	1	CH1	CH2	CH3	CH0	CH4	CH5	CH6	CH7
	CH1	1	1	CH2	CH3	CH0	CH1	CH4	CH5	CH6	CH7
	CH2	1	1	CH3	CH0	CH1	CH2	CH4	CH5	CH6	CH7
	CH3	1	1	CH0	CH1	CH2	CH3	CH4	CH5	CH6	CH7
	CH4	1	1	CH0	CH1	CH2	CH3	CH4	CH5	CH6	CH7
	CH5	1	1	CH0	CH1	CH2	CH3	CH4	CH5	CH6	CH7
	CH6	1	1	CH0	CH1	CH2	CH3	CH4	CH5	CH6	CH7
	CH7	1	1	CH0	CH1	CH2	CH3	CH4	CH5	CH6	CH7

11.2.13 DMA Operation Flag Bit Register (DMAFR)

DMAFR is an 8-bit readable/writable register that shows the DMA transfer status.

DMAFR is initialized to H'00 by a power-on reset or a transition to the hardware standby mode and retains its value in module standby mode.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	AE	-	-	-	NMIF
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/(W)*	R	R	R	R/(W)*

Note: * To clear the flag, write 0 to the bit to be cleared, only when 1 has been read from the bit when these registers have been read. To the AE or NMIF bit not to be cleared, only 1 can be written, even if 0 is read from the bit. To the reserved bits, only 0 can be written. When the flag is read by the CPU, even though 0 has been read, the flag may be set to 1 and therefore 1 may be read internally. In this case, if 0 is written to the corresponding bit, the flag will be cleared even if 1 has not been read by the CPU. Since the specifications of the flag bits in these registers differ from the ones of the flags in other modules, the clearing operations of the flags by the CPU must be executed carefully.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	AE	0	R/(W)*	Address Error Flag Indicates whether an address error has occurred by the DMAC. When this bit is set, even if the DE bit in CHCR and the DME bit in DMAOR are set to 1, DMA transfer is not enabled. This bit can only be cleared by writing 0 after reading as 1. The DMAC internal operation for an address error is as follows: • No address error: Read (source to DMAC) → Write (DMAC to destination) • Address error in source address: Nop → Nop • Address error in destination address: Read → Nop 0: No DMAC address error 1: DMAC address error occurred [Clearing condition] • Writing 0 after reading AE = 1

Bit	Bit Name	Initial Value	R/W	Description
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	NMIF	0	R/(W)*	NMI Flag Indicates that an NMI interrupt occurred. When this bit is set, even if the DE bit in CHCR and the DME bit in DMAOR are set to 1, DMA transfer is not enabled. This bit can only be cleared by writing 0 after reading as 1. When the NMI is input, the current DMA transfer in progress is continued until one transfer unit of data is completed. When DMAC operation is stopped by the NMI, it can be resumed by clearing the NMIF bit to 0 after reading as 1. Even if the NMI interrupt is input while the DMAC is not in operation, the NMIF bit is set to 1. 0: No NMI interrupt 1: NMI interrupt occurred [Clearing condition] • Writing 0 after reading NMIF = 1

Note: * Only 0 can be written to clear the flag after reading it as 1.

11.2.14 DMA Extension Resource Selectors 0 to 3 (DMARS0 to DMARS3)

The DMA extension resource selectors (DMARS) are 16-bit readable/writable registers that specify the DMA transfer sources from peripheral modules in each channel. DMARS0 is for channels 0 and 1, DMARS1 is for channels 2 and 3, DMARS2 is for channels 4 and 5, and DMARS3 is for channels 6 and 7. Table 11.3 shows the specifiable combinations.

DMARS can specify transfer requests from 36 ATU-III sources, 10 ADC sources (only for DMARS), and two CMT sources.

DMARS is initialized to H'0000 by a power-on reset or a transition to the hardware standby mode and retains its value in module standby mode.

- DMARS0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CH1 MID[5:0]						CH1 RID[1:0]		CH0 MID[5:0]						CH0 RID[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- DMARS1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CH3 MID[5:0]						CH3 RID[1:0]		CH2 MID[5:0]						CH2 RID[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- DMARS2

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CH5 MID[5:0]						CH5 RID[1:0]		CH4 MID[5:0]						CH4 RID[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- DMARS3

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CH7 MID[5:0]						CH7 RID[1:0]		CH6 MID[5:0]						CH6 RID[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Transfer requests from the various modules specify MID and RID as shown in table 11.3.

Table 11.3 DMARS Settings

Peripheral Module	Setting Value for One Channel ({MID, RID})	MID	RID	Function
ATU-III_C0	H'03	B'000000	B'11	—
ATU-III_C1	H'07	B'000001	B'11	—
ATU-III_C2	H'0B	B'000010	B'11	—
ATU-III_D00	H'13	B'000100	B'11	—
ATU-III_D01	H'17	B'000101	B'11	—
ATU-III_D02	H'1B	B'000110	B'11	—
ATU-III_D03	H'1F	B'000111	B'11	—
ATU-III_E0	H'23	B'001000	B'11	—
ATU-III_E1	H'27	B'001001	B'11	—
ATU-III_E2	H'2B	B'001010	B'11	—
ATU-III_E3	H'2F	B'001011	B'11	—
ATU-III_E4	H'33	B'001100	B'11	—
ATU-III_E5	H'37	B'001101	B'11	—
ATU-III_E6	H'A7	B'101001	B'11	—
ATU-III_C3	H'3B	B'001110	B'11	—
ATU-III_C4	H'3F	B'001111	B'11	—
ATU-III_G0	H'43	B'010000	B'11	—
ATU-III_G1	H'47	B'010001	B'11	—
ATU-III_G2	H'4B	B'010010	B'11	—
ATU-III_G3	H'4F	B'010011	B'11	—
ATU-III_D10	H'53	B'010100	B'11	—
ATU-III_D11	H'57	B'010101	B'11	—
ATU-III_D12	H'5B	B'010110	B'11	—
ATU-III_D13	H'5F	B'010111	B'11	—
ADC_B AN40	H'63	B'011000	B'11	—
ADC_B AN41	H'67	B'011001	B'11	—
ADC_B AN42	H'6B	B'011010	B'11	—
ADC_B AN43	H'6F	B'011011	B'11	—
ADC_B AN44	H'73	B'011100	B'11	—

Peripheral Module	Setting Value for One Channel ({MID, RID})	MID	RID	Function
ADC_B AN45	H'77	B'011101	B'11	—
ADC_B AN46	H'7B	B'011110	B'11	—
ADC_B AN47	H'7F	B'011111	B'11	—
ATU-III_D20	H'93	B'100100	B'11	—
ATU-III_D21	H'97	B'100101	B'11	—
ATU-III_D22	H'9B	B'100110	B'11	—
ATU-III_D23	H'9F	B'100111	B'11	—
ATU-III_D30	H'D3	B'110100	B'11	—
ATU-III_D31	H'D7	B'110101	B'11	—
ATU-III_D32	H'DB	B'110110	B'11	—
ATU-III_D33	H'DF	B'110111	B'11	—
ATU-III_J0	H'E3	B'111000	B'11	—
ATU-III_J1	H'E7	B'111001	B'11	—
ATU-III_G4	H'EF	B'111011	B'11	—
ATU-III_G5	H'F3	B'111100	B'11	—
CMT_0	H'FB	B'111110	B'11	—
CMT_1	H'FF	B'111111	B'11	—

When MID or RID other than the values listed in table 11.3 is set, the operation of this LSI is not guaranteed. The transfer request from DMARS is valid only when the resource select bits (RS3 to RS0) in CHCR0 to CHCR7 have been set to B'1000. Otherwise, even if DMARS has been set, the transfer request source is not accepted.

11.3 Operation

When a DMA transfer is requested, the DMAC starts the transfer according to the predetermined channel priority order; when the transfer end conditions are satisfied, it ends the transfer. Transfers can be requested in two modes: an auto request and an on-chip peripheral module request. In bus mode, the burst mode or the cycle stealing mode can be selected.

11.3.1 Transfer Flow

The DMA source address registers (SAR), DMA destination address registers (DAR), DMA transfer count registers (DMATCR), DMA channel control registers (CHCR), DMA operation register (DMAOR), and DMA extension resource selector (DMARS) are set for the target transfer conditions.

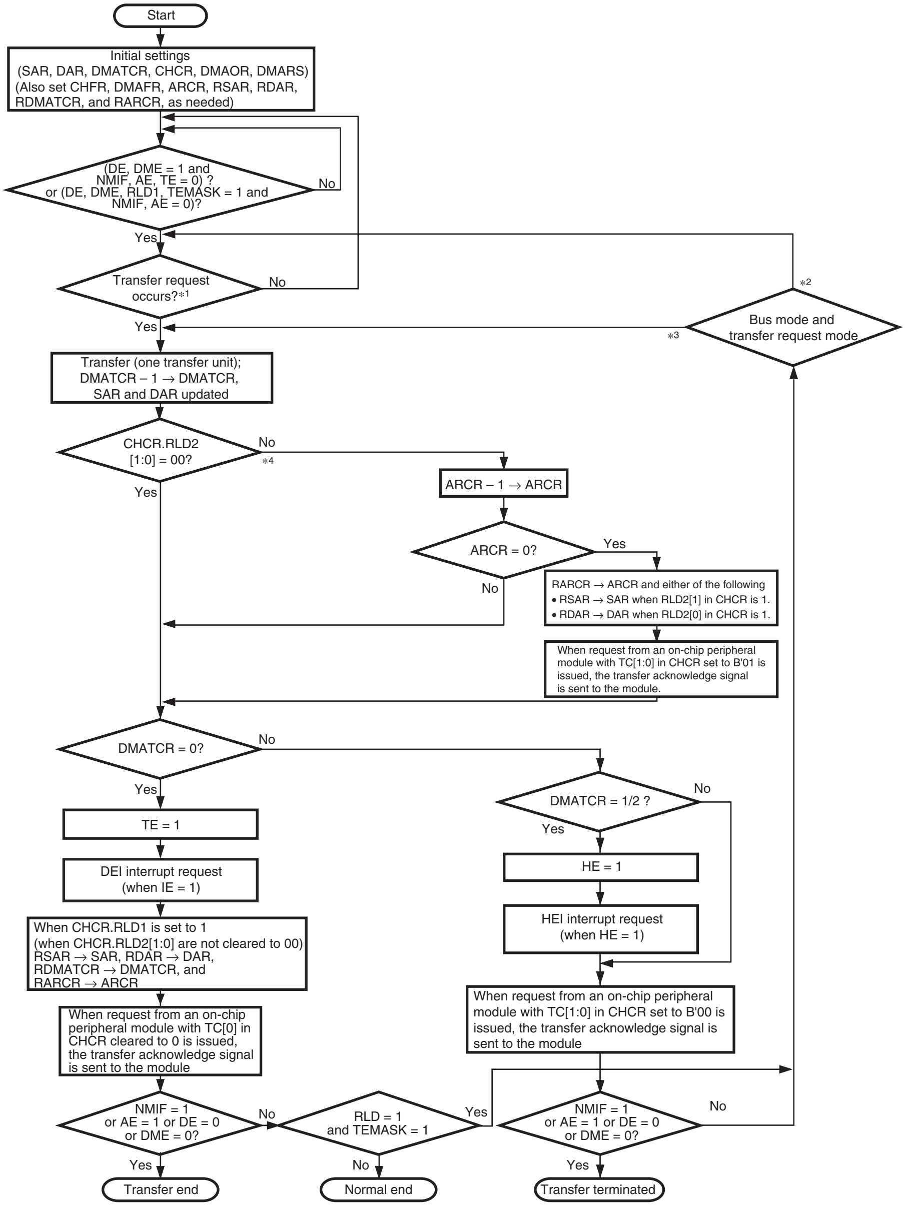
When the reloading function is used, the DMA address reload count register (ARCR), DMA reload-address reload count register (RARCR), DMA reload source address register (RSAR), DMA reload destination address register (RDAR), and DMA reload transfer count register (RDMATCR) are set as needed. If transfer has been started or an NMI interrupt or an address error has been generated, the DMA channel flag bit register (CHFR) and DMA operation flag bit register (DMAFR) are cleared as needed.

After the above registers are set, the DMAC transfers data according to the following procedure.

1. Checks to see if transfer is enabled (DE = 1, DME = 1, TE = 0, AE = 0, NMIF = 0)
2. When a transfer request comes and transfer is enabled, the DMAC transfers one transfer unit of data (depending on the TS1 and TS0 settings). For an auto request, the transfer begins automatically when the DE bit and DME bit are set to 1. The DMATCR value will be decremented by 1 for each transfer. If the RLD2[1:0] bits in CHCR is set to a value other than B'00, the ARCR value will also be decremented by 1. The actual transfer flows vary by address mode and bus mode.
3. If the TC[1:0] bits in CHCR are cleared to B'00 and the request is issued from an on-chip peripheral module, the transfer acknowledge signal will be returned to the module.
4. If the RLD2[1] and RLD0 bits in CHCR are set to a value other than B'00, the reloading function 2 is activated when transfers have been completed for the count specified by ARCR. When the RLD2[1] bit is set to 1, the reloading operations RSAR → SAR and RARCR → ARCR are performed. When the RLD2[0] bit is set to 1, the reloading operations RDAR → DAR and RARCR → ARCR are performed. If the TC[1:0] bits in CHCR are set to B'01 and the request is issued from an on-chip peripheral module, the transfer acknowledge signal will be returned to the module.

5. When half of the specified transfer count is exceeded (when DMATCR reaches half of the initial value), an HEI interrupt is sent to the CPU if the HE bit in CHFR is set to 1 while the HIE bit in CHCR is 1.
6. When transfer has been completed for the specified count (when DMATCR reaches 0), the transfer ends normally by setting the TE bit in CHFR to 1. If the IE bit in CHCR is set to 1 at this time, a DEI interrupt is sent to the CPU. If the RLD1 bit in CHCR is set to 1, the reloading function 1 is activated. The reloading operations RSAR → SAR, RDAR → DAR, RDMATCR → DMATCR, and RARCR → ARCR (only when the reloading function 2 is enabled) are performed. In addition, if RDL1 and TEMASK bits are both set to 1, the operation will be brought back to step 1 and transfer is kept performed until no transfer request is left to be handled. If the TC[0] bit in CHCR is set to 0 and the request is issued from an on-chip peripheral module, the transfer acknowledge signal will be returned to the module.
7. When an address error in the DMAC or an NMI interrupt is generated, the transfer is terminated. Transfers are also terminated when the DE bit in CHCR or the DME bit in DMAOR is cleared to 0.

Figure 11.2 is a flowchart of this procedure.


Start
Initial settings
(SAR, DAR, DMATCR, CHCR, DMAOR, DMARS)
(Also set CHFR, DMAFR, ARCR, RSAR, RDAR, RDMATCR, and RARCR, as needed)
(DE, DME = 1 and NMIF, AE, TE = 0) ? or (DE, DME, RLD1, TEMASK = 1 and NMIF, AE = 0)?
No
Yes
Transfer request occurs?*1
No
Yes
*2
Bus mode and transfer request mode
*3
Transfer (one transfer unit); DMATCR – 1 → DMATCR, SAR and DAR updated
CHCR.RLD2 [1:0] = 00?
No
*4
Yes
ARCR – 1 → ARCR
ARCR = 0?
Yes
No
RARCR → ARCR and either of the following
• RSAR → SAR when RLD2[1] in CHCR is 1.
• RDAR → DAR when RLD2[0] in CHCR is 1.
When request from an on-chip peripheral module with TC[1:0] in CHCR set to B'01 is issued, the transfer acknowledge signal is sent to the module.
DMATCR = 0?
No
Yes
TE = 1
DEI interrupt request (when IE = 1)
When CHCR.RLD1 is set to 1 (when CHCR.RLD2[1:0] are not cleared to 00) RSAR → SAR, RDAR → DAR, RDMATCR → DMATCR, and RARCR → ARCR
When request from an on-chip peripheral module with TC[0] in CHCR cleared to 0 is issued, the transfer acknowledge signal is sent to the module
DMATCR = 1/2 ?
No
Yes
HE = 1
HEI interrupt request (when HE = 1)
When request from an on-chip peripheral module with TC[1:0] in CHCR set to B'00 is issued, the transfer acknowledge signal is sent to the module
NMIF = 1 or AE = 1 or DE = 0 or DME = 0?
No
Yes
RLD = 1 and TEMASK = 1
Yes
No
NMIF = 1 or AE = 1 or DE = 0 or DME = 0?
No
Yes
Transfer end
Normal end
Transfer terminated

Notes: 1. In auto-request mode, if the TE bit is set to 0 or the RLD1 and TEMASK are both set to 1, and also the NMIF, AE, and TE bits are set to 0, setting the DE and DME bits to 1 initiates transfer.
2. Cycle stealing mode
3. Auto request mode in burst mode
4. Available only in channels 4 to 7. In channels 0 to 3, RLD2[1] and RLD2[0] in CHCR are always assumed to be 00.

Figure 11.2 DMA Transfer Flowchart

11.3.2 DMA Transfer Requests

DMA transfer requests are basically generated in either the data transfer source or destination, but they can also be generated in external devices and on-chip peripheral modules that are neither the transfer source nor destination.

Transfers can be requested in two modes: an auto request and an on-chip peripheral module request. The request mode is selected by the RS3 to RS0 bits in CHCR0 to CHCR7 and DMARS0 to DMARS3.

(1) Auto-Request Mode

When there is no transfer request signal from an external source, as in a memory-to-memory transfer or a transfer between memory and an on-chip peripheral module unable to request a transfer, the auto-request mode allows the DMAC to automatically generate a transfer request signal internally. When the DE bits in CHCR0 to CHCR7 and the DME bit in DMAOR are set to 1, the transfer begins so long as the TE bits in CHFR0 to CHFR7, and the AE and NMIF bits in DMAFR are 0.

(2) On-Chip Peripheral Module Request

In this mode, the transfer is performed in response to the DMA transfer request signal from an on-chip peripheral module.

Signals that request DMA transfer from on-chip peripheral modules include A/D conversion end transfer requests from the ADC, transfer requests from the RCAN-TL1 or ATU-III, and compare-match transfer requests from the CMT.

When a transfer request signal is sent in on-chip peripheral module request mode while DMA transfer is enabled (DE = 1, DME = 1, TE = 0, AE = 0, and NMIF = 0), DMA transfer is performed.

When a transfer is requested from the ADC, the transfer source must be the A/D data register (ADDR). Any address can be specified for data transfer source and destination when a transfer request is generated by the CMT or ATU-III.

Table 11.4 Selecting On-Chip Peripheral Module Request Modes with RS3 to RS0 Bits

CHCR	DMARS		DMA Transfer				
RS[3:0]	MID	RID	Request Source	DMA Transfer Request Signal	Transfer Source	Transfer Destination	Bus Mode
1001	Any	Any	RCAN-TL1-A	RMA0 (reception complete)	RCAN0 (MB0)	Any	Cycle stealing
1010	Any	Any	RCAN-TL1-B	RMB0 (reception complete)	RCAN1 (MB0)	Any	
1011	Any	Any	RCAN-TL1-C	RMC0 (reception complete)	RCAN2 (MB0)	Any	
1100	Any	Any	ADC_A	ADI0 (scan conversion end)	ADR0 to ADR39	Any	Cycle stealing/ burst
1101	Any	Any	ADC_B	ADI1 (scan conversion end)	ADR40 to ADR47	Any	
1110	Any	Any	RCAN-TL1 channel D	RMD0 (reception complete)	RCAN3 (MB0)	Any	Cycle stealing
1000	000000	11	ATU-III_C0	IMIC00 (input capture/compare match)	Any	Any	Cycle stealing/ burst
	000001	11	ATU-III_C1	IMIC10 (input capture/compare match)	Any	Any	
	000010	11	ATU-III_C2	IMIC20 (input capture/compare match)	Any	Any	
	001110	11	ATU-III_C3	IMIC30 (input capture/compare match)	Any	Any	
	001111	11	ATU-III_C4	IMIC40 (input capture/compare match)	Any	Any	
	000100	11	ATU-III_D00	UDID00 (down-counter underflow)	Any	Any	Cycle stealing/ burst
	000101	11	ATU-III_D01	UDID01 (down-counter underflow)	Any	Any	
	000110	11	ATU-III_D02	UDID02 (down-counter underflow)	Any	Any	

CHCR	DMARS		DMA Transfer				
RS3 to RS0	MID	RID	Request Source	DMA Transfer Request Signal	Transfer Source	Transfer Destination	Bus Mode
1000	000111	11	ATU-III_D03	UDID03 (down-counter underflow)	Any	Any	Cycle stealing/ burst
	001000	11	ATU-III_E0	CMIE00 (compare match)	Any	Any	
	001001	11	ATU-III_E1	CMIE10 (compare match)	Any	Any	
	001010	11	ATU-III_E2	CMIE20 (compare match)	Any	Any	
	001011	11	ATU-III_E3	CMIE30 (compare match)	Any	Any	
	001100	11	ATU-III_E4	CMIE40 (compare match)	Any	Any	
	001101	11	ATU-III_E5	CMIE50 (compare match)	Any	Any	
	101001	11	ATU-III_E6	CMIE60 (compare match)	Any	Any	
	010000	11	ATU-III_G0	CMIG0 (compare match)	Any	Any	Cycle stealing/ burst
	010001	11	ATU-III_G1	CMIG2 (compare match)	Any	Any	
	010010	11	ATU-III_G2	CMIG3 (compare match)	Any	Any	
	010011	11	ATU-III_G3	CMIG4 (compare match)	Any	Any	
	010100	11	ATU-III_D10	UDID10 (down-counter underflow)	Any	Any	Cycle stealing/ burst
	010101	11	ATU-III_D11	UDID11 (down-counter underflow)	Any	Any	
	010110	11	ATU-III_D12	UDID12 (down-counter underflow)	Any	Any	
	010111	11	ATU-III_D13	UDID13 (down-counter underflow)	Any	Any	

CHCR	DMARS		DMA Transfer				
RS3 to RS0	MID	RID	Request Source	DMA Transfer Request Signal	Transfer Source	Transfer Destination	Bus Mode
1000	011000	11	ADC_B AN40	ADID40 (interrupt conversion end)	ADR40	Any	Cycle stealing
	011001	11	ADC_B AN 41	ADID41 (interrupt conversion end)	ADR41	Any	
	011010	11	ADC_B AN 42	ADID42 (interrupt conversion end)	ADR42	Any	
	011011	11	ADC_B AN 43	ADID43 (interrupt conversion end)	ADR43	Any	
	011100	11	ADC_B AN 44	ADID44 (interrupt conversion end)	ADR44	Any	
	011101	11	ADC_B AN 45	ADID45 (interrupt conversion end)	ADR45	Any	
	011110	11	ADC_B AN 46	ADID46 (interrupt conversion end)	ADR46	Any	
	011111	11	ADC_B AN 47	ADID47 (interrupt conversion end)	ADR47	Any	
	100100	11	ATU-III_D20	UDID20 (down-counter underflow)	Any	Any	Cycle stealing/ burst
	100101	11	ATU-III_D21	UDID21 (down-counter underflow)	Any	Any	
	100110	11	ATU-III_D22	UDID22 (down-counter underflow)	Any	Any	
	100111	11	ATU-III_D23	UDID23 (down-counter underflow)	Any	Any	
	110100	11	ATU-III_D30	UDID30 (down-counter underflow)	Any	Any	Cycle stealing/ burst
	110101	11	ATU-III_D31	UDID31 (down-counter underflow)	Any	Any	

CHCR	DMARS		DMA Transfer				
RS3 to RS0	MID	RID	Request Source	DMA Transfer Request Signal	Transfer Source	Transfer Destination	Bus Mode
1000	110110	11	ATU-III_D32	UDID32 (down-counter underflow)	Any	Any	Cycle stealing/ burst
	110111	11	ATU-III_D33	UDID33 (down-counter underflow)	Any	Any	
	111000	11	ATU-III_J0	DFIJ0 (FIFO full)	Any	Any	
	111001	11	ATU-III_J1	DFIJ1 (FIFO full)	Any	Any	
	111011	11	ATU-III_G4	CMIG4 (compare match)	Any	Any	
	111100	11	ATU-III_G5	CMIG5 (compare match)	Any	Any	
	111110	11	CMT_0	CMI0 (compare match)	Any	Any	
	111111	11	CMT_1	CMI1 (compare match)	Any	Any	

Note: Set the TC[1:0] bits in CHCR to B'00 only when a DMA transfer request source for which any transfer source and destination can be set is used. Otherwise, operation is not guaranteed. For ADC_A and ADC_B, a value other than B'00 can be set.

11.3.3 Channel Priority

When the DMAC receives simultaneous transfer requests on two or more channels, it selects a channel according to a predetermined priority order. Three modes (fixed mode 1, fixed mode 2, and round-robin mode) are selected using the PR[1:0] bits in DMAOR.

(1) Fixed Mode

In fixed modes, the priority levels among the channels remain fixed. There are two kinds of fixed modes as follows:

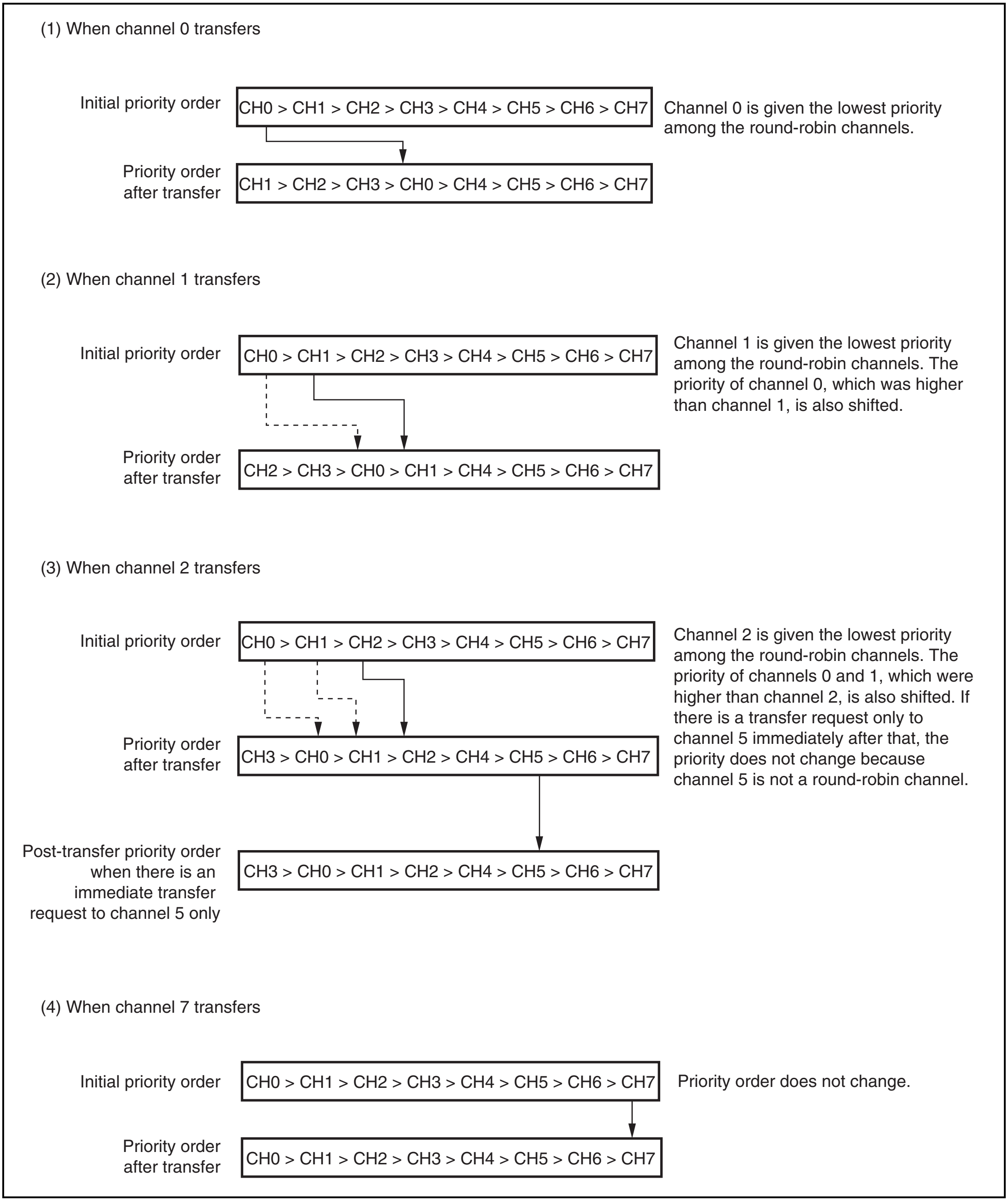
- Fixed mode 1: CH0 > CH1 > CH2 > CH3 > CH4 > CH5 > CH6 > CH7
- Fixed mode 2: CH0 > CH4 > CH1 > CH5 > CH2 > CH6 > CH3 > CH7

These are selected by the PR[1:0] bits in the DMA operation register (DMAOR).

(2) Round-Robin Mode

Each time one unit of word, byte, longword, or 16 bytes is transferred on one channel, the priority order is rotated. The channel on which the transfer was just finished is rotated to the lowest of the priority order among the four round-robin channels (channels 0 to 4). The priority of the channels other than the round-robin channels (channels 0 to 4) does not change even in round-robin mode. The round-robin mode operation is shown in figure 11.3. The priority in round-robin mode is CH0 > CH1 > CH2 > CH3 > CH4 > CH5 > CH6 > CH7 immediately after a reset.

When the round-robin mode has been specified, do not concurrently specify cycle stealing mode and burst mode as the bus modes of any two or more channels.


(1) When channel 0 transfers
Initial priority order
CH0 > CH1 > CH2 > CH3 > CH4 > CH5 > CH6 > CH7
Priority order after transfer
CH1 > CH2 > CH3 > CH0 > CH4 > CH5 > CH6 > CH7
(2) When channel 1 transfers
Initial priority order
CH0 > CH1 > CH2 > CH3 > CH4 > CH5 > CH6 > CH7
Priority order after transfer
CH2 > CH3 > CH0 > CH1 > CH4 > CH5 > CH6 > CH7
(3) When channel 2 transfers
Initial priority order
CH0 > CH1 > CH2 > CH3 > CH4 > CH5 > CH6 > CH7
Priority order after transfer
CH3 > CH0 > CH1 > CH2 > CH4 > CH5 > CH6 > CH7
Post-transfer priority order when there is an immediate transfer request to channel 5 only
CH3 > CH0 > CH1 > CH2 > CH4 > CH5 > CH6 > CH7
(4) When channel 7 transfers
Initial priority order
CH0 > CH1 > CH2 > CH3 > CH4 > CH5 > CH6 > CH7
Priority order after transfer
CH0 > CH1 > CH2 > CH3 > CH4 > CH5 > CH6 > CH7

Channel 0 is given the lowest priority among the round-robin channels.

Channel 1 is given the lowest priority among the round-robin channels. The priority of channel 0, which was higher than channel 1, is also shifted.

Channel 2 is given the lowest priority among the round-robin channels. The priority of channels 0 and 1, which were higher than channel 2, is also shifted. If there is a transfer request only to channel 5 immediately after that, the priority does not change because channel 5 is not a round-robin channel.

Priority order does not change.

Figure 11.3 Round-Robin Mode

Figure 11.4 shows how the priority order changes when channel 0 and channel 3 transfers are requested simultaneously and a channel 1 transfer is requested during the channel 0 transfer. The DMAC operates as follows:

1. Transfer requests are generated simultaneously to channels 0 and 3.
2. Channel 0 has a higher priority, so the channel 0 transfer begins first (channel 3 waits for transfer).
3. A channel 1 transfer request occurs during the channel 0 transfer (channels 1 and 3 are both waiting)
4. When the channel 0 transfer ends, channel 0 is given the lowest priority among the round-robin channels.
5. At this point, channel 1 has a higher priority than channel 3, so the channel 1 transfer begins (channel 3 waits for transfer).
6. When the channel 1 transfer ends, channel 1 is given the lowest priority among the round-robin channels.
7. The channel 3 transfer begins.
8. When the channel 3 transfer ends, channels 3 and 2 are lowered in priority so that channel 3 is given the lowest priority among the round-robin channels.

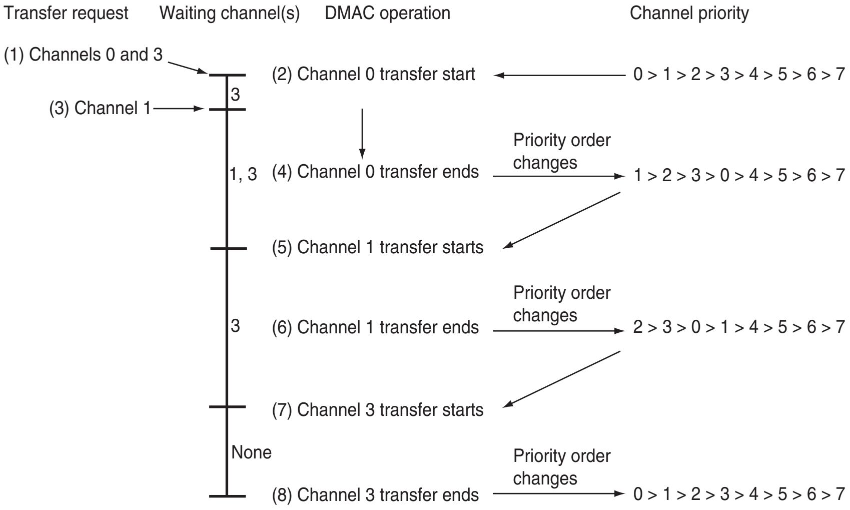


Figure 11.4 Changes in Channel Priority in Round-Robin Mode

11.3.4 DMA Transfer Types

This LSI supports DMA transfer in dual address mode. A data transfer timing depends on the bus mode, which is the cycle stealing mode or burst mode. The DMAC supports the transfers shown in table 11.5.

Table 11.5 Supported DMA Transfers

	Transfer Destination			
Transfer Source	**External Memory**	**External Device*[3]**	**On-Chip Peripheral Module**	**On-Chip Memory**
External memory	Dual	Dual	Dual	Dual
External device*[3]	Dual	Dual	Dual	Dual
On-chip peripheral module	Dual	Dual	Dual	Dual
On-chip memory	Dual	Dual	Dual	Dual

Notes: 1. Dual: Dual address mode

2. 16-byte transfer is available only for on-chip peripheral modules that support longword access.

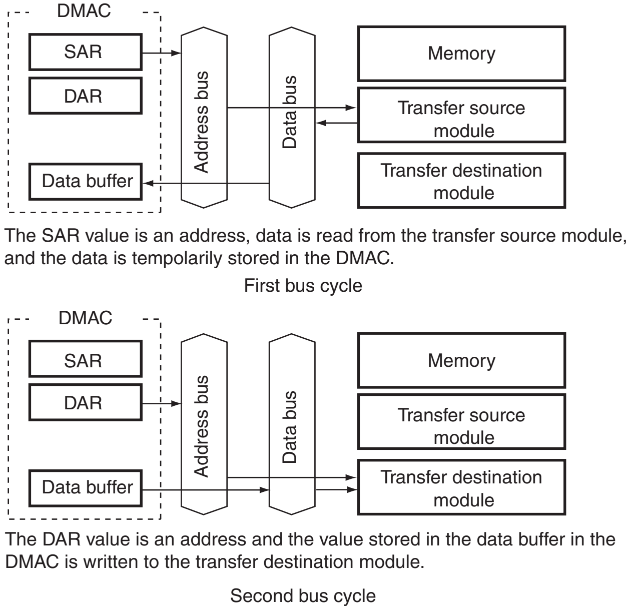
3. The external devices are the ones that are connected to the external address space.

(1) Address Modes

- Dual Address Mode

  In dual address mode, both the transfer source and destination are accessed (selected) by an address. The transfer source and destination can be located externally or internally.

  DMA transfer requires two bus cycles because data is read from the transfer source in a data read cycle and written to the transfer destination in a data write cycle. At this time, transfer data is temporarily stored in the DMAC.



The SAR value is an address, data is read from the transfer source module, and the data is tempolarily stored in the DMAC.

First bus cycle

The DAR value is an address and the value stored in the data buffer in the DMAC is written to the transfer destination module.

Second bus cycle

Figure 11.5 Data Flow of Dual Address Mode

Auto request and on-chip peripheral module request are available for the transfer request.

Figure 11.6 shows an example of DMA transfer timing in dual address mode.

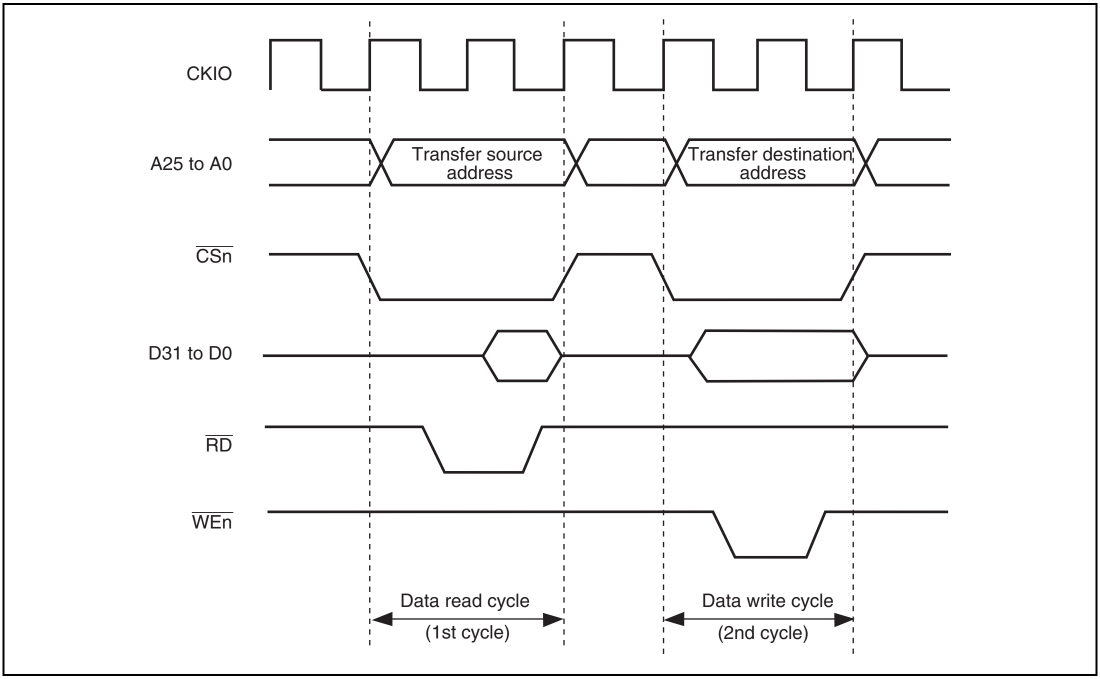


Figure 11.6 Example of DMA Transfer Timing in Dual Mode (Transfer Source: External Memory, Transfer Destination: External Memory)

(2) Bus Modes

There are two bus modes; cycle stealing and burst. Select the mode by the TB bits in the channel control registers (CHCR).

- Cycle Stealing Mode

— Normal mode

In normal mode of cycle stealing, the bus mastership is given to another bus master after a one-transfer-unit (byte, word, longword, or 16-byte unit) DMA transfer. When another transfer request occurs, the bus mastership is obtained from another bus master and a transfer is performed for one transfer unit. When that transfer ends, the bus mastership is passed to another bus master. This is repeated until the transfer end conditions are satisfied.

The cycle-stealing normal mode can be used for any transfer section; transfer request source, transfer source, and transfer destination.

Figure 11.7 shows an example of DMA transfer timing in cycle-stealing normal mode. Dual address mode transfer is performed.

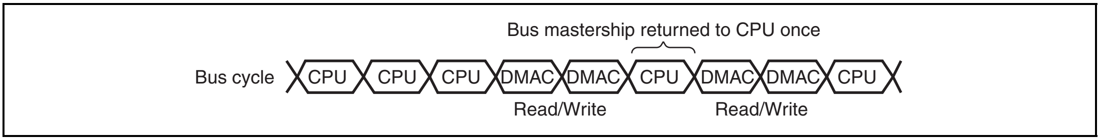


Figure 11.7 DMA Transfer Example in Cycle-Stealing Normal Mode (Dual Address)

— Intermittent Mode 16 and Intermittent Mode 64

In intermittent mode of cycle stealing, DMAC returns the bus mastership to other bus master whenever a unit of transfer (byte, word, longword, or 16 bytes) is completed. If the next transfer request occurs after that, DMAC obtains the bus mastership from other bus master after waiting 16 or 64 cycles of Pϕ clock. DMAC then transfers data of one unit and returns the bus mastership to other bus master. These operations are repeated until the transfer end condition is satisfied. It is thus possible to make lower the ratio of bus occupation by DMA transfer than the normal mode of cycle stealing.

When DMAC obtains again the bus mastership, DMA transfer may be postponed in case of entry updating due to cache miss.

The cycle-stealing intermittent mode can be used for any transfer section; transfer request source, transfer source, and transfer destination. The bus modes, however, must be cycle stealing mode in all channels.

Figure 11.8 shows an example of DMA transfer timing in cycle-stealing intermittent mode. Dual address mode transfer is performed.

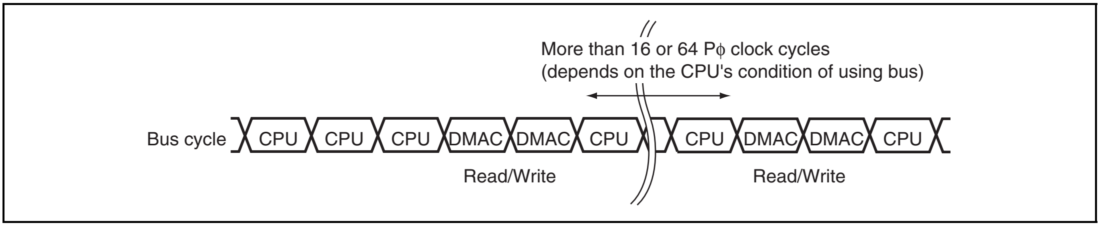

More than 16 or 64 Pϕ clock cycles
(depends on the CPU's condition of using bus)
Bus cycle
CPU
CPU
CPU
DMAC
DMAC
CPU
CPU
DMAC
DMAC
CPU
Read/Write
Read/Write

Figure 11.8 Example of DMA Transfer in Cycle-stealing intermittent Mode (Dual Address)

- Burst Mode

In burst mode, once the DMAC obtains the bus mastership, it does not release the bus mastership and continues to perform transfer until the transfer end condition is satisfied.

Figure 11.9 shows DMA transfer timing in burst mode.

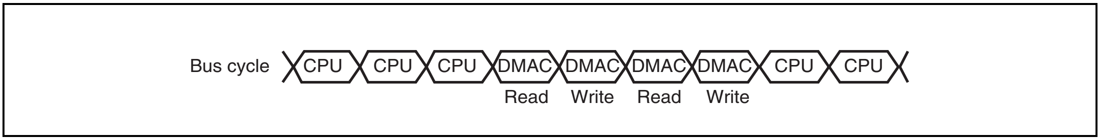


Figure 11.9 DMA Transfer Example in Burst Mode (Dual Address)

(3) Relationship between Request Modes and Bus Modes by DMA Transfer Category

Table 11.6 shows the relationship between request modes and bus modes by DMA transfer category.

Table 11.6 Relationship of Request Modes and Bus Modes by DMA Transfer Category

Address Mode	Transfer Category	Request Mode	Bus Mode	Transfer Size (Bits)	Usable Channels
Dual	External memory and external memory	All*[3]	B/C	8/16/32/128	0 to 7
	External memory and external device*[5]	All*[3]	B/C	8/16/32/128	0 to 7
	External device*[5] and external device*[5]	All*[3]	B/C	8/16/32/128	0 to 7
	External memory and on-chip peripheral module	All*[1]	B/C*[4]	8/16/32/128*[2]	0 to 7
	External device*[5] and on-chip peripheral module	All*[1]	B/C*[4]	8/16/32/128*[2]	0 to 7
	On-chip peripheral module and on-chip peripheral module	All*[1]	B/C*[4]	8/16/32/128*[2]	0 to 7
	On-chip memory and on-chip memory	All*[3]	B/C	8/16/32/128	0 to 7
	On-chip memory and external device*[5]	All*[3]	B/C	8/16/32/128	0 to 7
	On-chip memory and on-chip peripheral module	All*[1]	B/C*[4]	8/16/32/128*[2]	0 to 7
	On-chip memory and external memory	All*[3]	B/C	8/16/32/128	0 to 7

[Legend]

B: Burst

C: Cycle stealing

Notes: 1. Auto requests and on-chip peripheral module requests are both available. If the request is issued from an on-chip peripheral module, however, the register of the requesting on-chip peripheral module must be the transfer source or destination, other than when the transfer request source is the CMT or ATU-III.

2. Access size permitted for the on-chip peripheral module register functioning as the transfer source or transfer destination.

3. Auto requests and on-chip peripheral module requests are both available. For on-chip peripheral module requests, however, only the CMT and ATU-III are available as the transfer request source.

4. If the request is issued from an on-chip peripheral module, only cycle stealing mode transfer is available other than when the transfer request source is the CMT, ATU-III, 12-bit ADC_A or 12-bit ADC_B.

5. The external devices are the ones that are connected to the external address space.

(4) Bus Mode and Channel Priority

In priority fixed mode (CH0 > CH1), when channel 1 is transferring data in burst mode and a request arrives for transfer on channel 0, which has higher-priority, the data transfer on channel 0 will begin immediately. In this case, if the transfer on channel 0 is also in burst mode, the transfer on channel 1 will only resume on completion of the transfer on channel 0.

When channel 0 is in cycle stealing mode, one transfer-unit of data on this channel, which has the higher priority, is transferred. Data is then transferred continuously to channel 1 without releasing the bus. The bus mastership will then switch between the two in this order: channel 0, channel 1, channel 0, channel 1, etc. That is, the CPU cycle after the data transfer in cycle stealing mode is replaced with a burst-mode transfer cycle (priority execution of burst-mode cycle). An example of this is shown in figure 11.10.

When multiple channels are in burst mode, data transfer on the channel that has the highest priority is given precedence. When DMA transfer is being performed on multiple channels, the bus mastership is not released to another bus-master device until all of the competing burst-mode transfers have been completed.

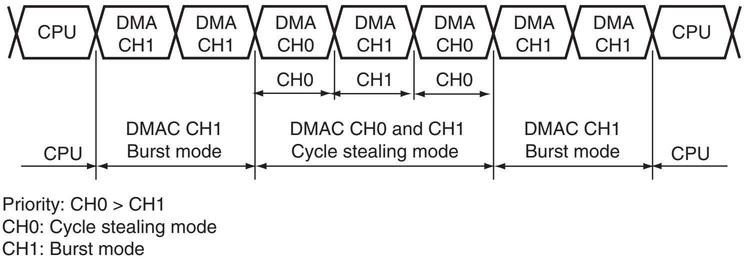


Figure 11.10 Bus State when Multiple Channels are Operating

In round-robin mode, the priority changes as shown in figure 11.4. Note that channels in cycle stealing and burst modes must not be mixed.

11.4 Special Operations

The special operations of the DMAC are described below.

11.4.1 Address Error Operation

As explained in the description of the address flag bit (AE) in section 11.2.13, DMA Operation Flag Bit Register (DMAFR), if an address error occurs during transfer operations on the DMAC, the following operation is performed:

- No address error occurred: read (source → DMAC internal) → write (DMAC internal → destination)
- An address error occurred in the source address: Nop → Nop
- An address error occurred in the destination address: read → Nop

11.4.2 Operation on NMI

As explained in the description of the NMI flag bit (NMIF) in section 11.2.13, DMA Operation Flag Bit Register (DMAFR), if an NMI occurs during a DMA transfer, the current transfer in progress is continued until one transfer unit of data is completed.

In this case, the DMAC operation can be resumed by clearing the NMIF bit to 0 after reading as 1. The operation can be resumed regardless of whether the reloading functions are enabled or disabled.

11.4.3 Operation of Reloading Function 1

Figure 11.11 shows the operation that takes place when the reloading function 1 is enabled and when the reloading function 2 is enabled, as well as the assert timing for the transfer acknowledge signal to an on-chip peripheral module.

Figure 11.12 shows addresses, commands, and data in the internal bus, the states of the peripheral bus, and the update status of the various registers.

Note: Even if the reloading function 1 is enabled, data is not reloaded from RARCR → ARCR. when the reloading function 2 is disabled. The reloading operation is performed only when the reloading functions 1 and 2 are both enabled.

11.4.4 Operation of Reloading Function 2

Figure 11.11 shows the operation that takes place when the reloading function 2 is enabled and when the reloading function 1 is enabled, as well as the assert timing for the transfer acknowledge signal to an on-chip peripheral module.

11.4.5 Interface with On-Chip Peripheral Module

Figure 11.11 shows the assert timing of the transfer acknowledge signal to an on-chip peripheral module and when the reloading function 1 is enabled and in conjunction with the operation that takes place when the reloading function 2 is enabled.

The assertion timing of the acknowledge signal depends on the settings of the TC[1:0] bits in CHCR.

In channels 0 to 3, the TC[0] bit is always fixed at 0. In channels 4 to 7, setting the TC[1:0] bits to B'11 is prohibited. Further, if the TC[1:0] bits are to be set to B'01, either the RLD2[1] bit or the RLD2[0] bit in CHCR, or both, should be set to 1. If the RLD2[1:0] bits are B'00, the integrity of the resulting operation cannot be guaranteed.

11.4.6 Integer to Floating-Point Conversion Operation

As explained in the description of the integer to floating-point function enable bit (IFT) in section 11.2.4, DMA Channel Control Registers 0 to 7 (CHCR0 to CHCR7), when the integer to floating-point conversion function is enabled, the following conversion operations are required:

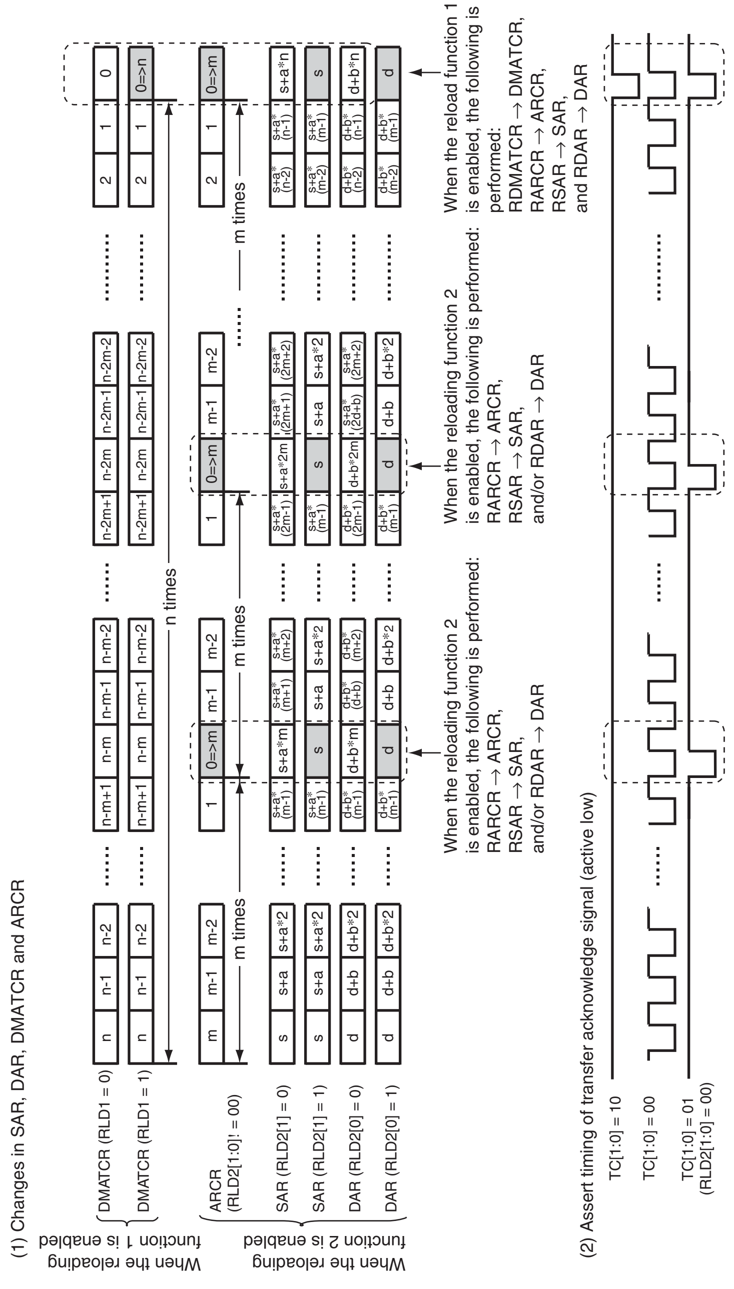
- No conversion required: read (transfer source to DMAC internal) → write (DMAC internal to transfer destination)
- A conversion required: read → Nop (conversion) → write

The conversion processing involves unusual operations of reading 16-bit data from the transfer source and writing 32-bit data to the transfer destination. For this reason, the TS[1:0] bits in CHCR should be set to B'01 to match the data size at the source of transfer. If the bits are set to any other value, the resulting operation cannot be guaranteed. Further, the transfer destination addresses should be aligned with a 4-byte boundary.

Table 11.7 shows conversion examples on several values. The LSB and the bit next to the LSB of data to be conversion are not converted. The decimal point is indicated by "." in table 11.7.

Table 11.7 Integer to Floating-Point Conversion Example

Data to be Converted (Hexadecimal)	Converted Data (Hexadecimal)
.0000 0000 0000 00 00 (.0000)	0000 0000 0000 0000 0000 0000 00000000 (00000000)
.0000 0000 0000 01 00 (.0004)	0011 1000 1000 0000 0000 0000 00000000 (38800000)
.0000 0000 0000 10 00 (.0008)	0011 1001 0000 0000 0000 0000 00000000 (39000000)
.0100 0000 0000 00 00 (.4000)	0011 1110 1000 0000 0000 0000 00000000 (3E800000)
.1000 0000 0000 00 00 (.8000)	0011 1111 0000 0000 0000 0000 00000000 (3F000000)
.0101 0101 0101 01 00 (.5554)	0011 1110 1010 1010 1010 1000 00000000 (3EAAA800)
.1010 1010 1010 10 00 (.AAA8)	0011 1111 0010 1010 1010 1000 00000000 (3F2AA800)
.1100 1100 1100 11 00 (.CCCC)	0011 1111 0100 1100 1100 1100 00000000 (3F4CCC00)
.0011 0011 0011 00 00 (.3330)	0011 1110 0100 1100 1100 0000 00000000 (3F4CC000)
.1110 0011 1000 11 00 (.E38C)	0011 1111 0110 0011 1000 1100 00000000 (3F638C00)
.0001 1100 0111 00 00 (.1C70)	0011 1101 1110 0011 1000 0000 00000000 (3DE38000)
.1111 0000 1111 00 00 (.F0F0)	0011 1111 0111 0000 1111 0000 00000000 (3F70F000)
.0000 1111 0000 11 00 (.0F03)	0011 1101 0111 0000 0011 0000 00000000 (3D703000)
.1111 1111 1111 00 00 (.FFF0)	0011 1111 0111 1111 1111 0000 00000000 (3F7FF000)


(1) Changes in SAR, DAR, DMATCR and ARCR
When the reloading function 1 is enabled
DMATCR (RLD1 = 0)
DMATCR (RLD1 = 1)
When the reloading function 2 is enabled
ARCR (RLD2[1:0]! = 00)
SAR (RLD2[1] = 0)
SAR (RLD2[1] = 1)
DAR (RLD2[0] = 0)
DAR (RLD2[0] = 1)
n
n-1
n-2
n-m+1
n-m
n-m-1
n-m-2
n-2m+1
n-2m
n-2m-1
n-2m-2
2
1
0
0=>n
n times
m
m-1
m-2
1
0=>m
m times
s
s+a
s+a*2
s+a*(m-1)
s+a*m
s+a*(m+1)
s+a*(m+2)
s+a*(2m-1)
s+a*2m
s+a*(2m+1)
s+a*(2m+2)
s+a*(n-2)
s+a*(n-1)
s+a*n
s+a*(m-2)
d
d+b
d+b*2
d+b*(m-1)
d+b*m
d+b*(d+b)
d+b*(m+2)
d+b*(2m-1)
d+b*2m
s+a*(2d+b)
s+a*(2m+2)
d+b*(n-2)
d+b*(n-1)
d+b*n
d+b*(m-2)
When the reloading function 2 is enabled, the following is performed: RARCR → ARCR, RSAR → SAR, and/or RDAR → DAR
When the reloading function 2 is enabled, the following is performed: RARCR → ARCR, RSAR → SAR, and/or RDAR → DAR
When the reload function 1 is enabled, the following is performed: RDMATCR → DMATCR, RARCR → ARCR, RSAR → SAR, and RDAR → DAR
(2) Assert timing of transfer acknowledge signal (active low)
TC[1:0] = 10
TC[1:0] = 00
TC[1:0] = 01 (RLD2[1:0] = 00)

[Legend]

n: Transfer count specified in DMATCR and RDMATCR. Any integer can be set ($\leq 2^{24}$).

m: Reload count specified in ARCR and RARCR. Any integer can be set ($\leq 2^{8}$).

0 => n: Reaches 0 but set to n by the reloading function

s: Address specified in SAR and RSAR

d: Address specified in DAR and RDAR

a: Difference when SAR is updated (±1 for byte, ±2 for word, or ±4 for longword)

b: Difference when DAR is updated (±1 for byte, ±2 for word, or ±4 for longword)

Figure 11.11 Reloading Functions and Timing of Transfer Acknowledge Signal to On-Chip Peripheral Module

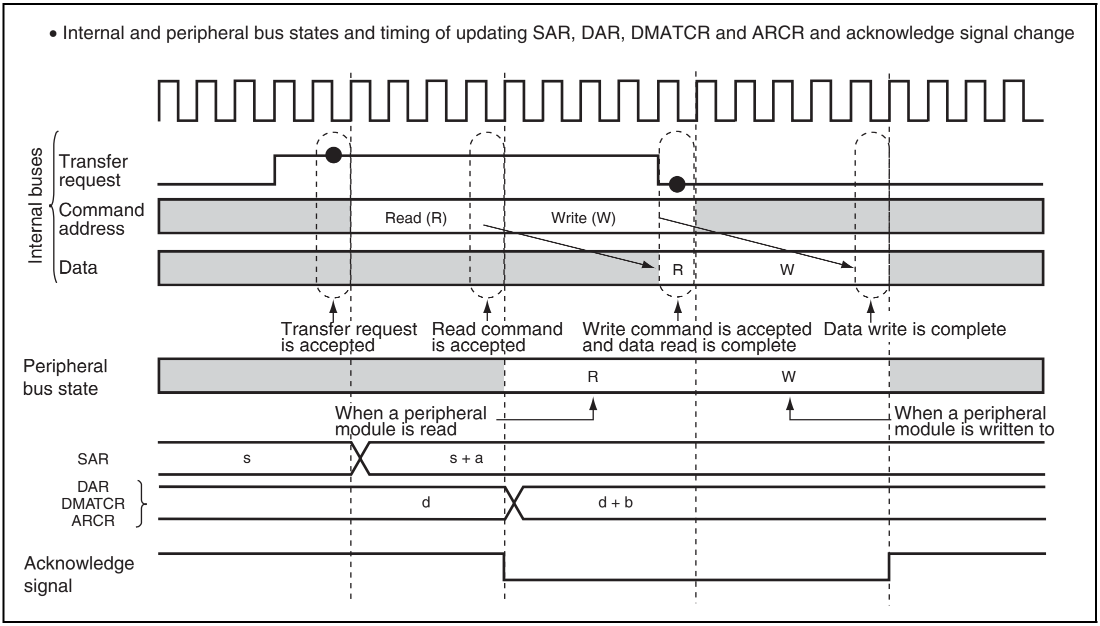

• Internal and peripheral bus states and timing of updating SAR, DAR, DMATCR and ARCR and acknowledge signal change
Internal buses
Transfer request
Command address
Data
Read (R)
Write (W)
R
W
Transfer request is accepted
Read command is accepted
Write command is accepted and data read is complete
Data write is complete
Peripheral bus state
R
W
When a peripheral module is read
When a peripheral module is written to
SAR
s
s + a
DAR
DMATCR
ARCR
d
d + b
Acknowledge signal

Figure 11.12 Change Timing of Register Contents and Acknowledge Signal

11.5 Usage Note

In CHFR0 to CHFR7 and DMAFR, when the flag is read by the CPU, even though 0 has been read, the flag may be set to 1 and therefore 1 may be read internally.

In this case, if 0 is written to the corresponding bit, the flag will be cleared even if 1 has not been read by the CPU.

To avoid the malfunction, the clear conditions must be observed, described in the notes of CHFR0 to CHFR7 and DMAFR.

Since the specifications of the flag bits in CHFR0 to CHFR7 and DMAFR differ from the ones of the flags in other modules, the clearing operations of the flags by the CPU must be executed carefully.

Section 12 Automotive Direct Memory Access Controller (A-DMAC)

The automotive direct memory access controller (A-DMAC) can be used in place of the CPU to perform high-speed data transfers between on-chip peripheral modules and on-chip RAM. The A-DMAC reduces the load of the CPU and improves this LSI operating efficiency.

12.1 Features

- Basic functions

  Performs high-speed data transfers between on-chip peripheral modules and on-chip RAM in place of the CPU.

  Transfer source and destination registers in on-chip peripheral modules are fixed in each channel.
- On-chip peripheral modules

  ADC, ATU-III (Timers A, C, F, and G), RSPI, SCI, RCAN-TL1 (hereinafter abbreviated as RCAN), FlexRay.
- Alias areas

  Areas in on-chip RAM which are used as transfer source and destination areas during data transfer by the A-DMAC are called alias areas.

  The start address of an alias area is specified as the start address (H'FFF80000) of the on-chip RAM as default. The start address of the alias area can be changed. The offset from the start address of the alias area for each channel can be specified arbitrarily by address registers for ATU-III (timer G), ADC, RSPI, and SCI channels. The offsets from the start address of the alias area for ATU-III (timers A, C, and F), RCAN, and FlexRay channels are fixed.
- Number of channels: 78
- Transfer requests: fixed for each A-DMAC channel
  - Channel for ADC: ADC (ADC_A, AN0) interrupt conversion end
  - Channels for ATU-III (Timers A, C, and F): Input capture
  - Channel for ATU-III (Timer G): Compare match
  - Channels for RSPI: Receive buffer full or transmit buffer empty
  - Channels for SCI: Receive data full or transmit data empty
  - Channels for RCAN (reception): Mailbox full
  - Channels for RCAN (transmission): Startup by software
  - Channels for FlexRay: Receive message buffer full or receive FIFO message buffer full

- A-DMAC channel functions
  - Channel for ADC: Performs ring-buffer type transfer from AN0 of ADC to on-chip RAM.
  - Channel for ATU-III (Timer G): Performs ring-buffer type transfer from on-chip RAM to port G.
  - Channels for ATU-III (Timers A, C, and F): Support register reading. Channels for timer A perform input capture register read and transfer the register read value and previous value buffered in the A-DMAC to the alias area. Channels for timers C and F transfer data from input capture register to the alias area. In the channels for timer F, either two register transfer (PWM input waveform measurement mode) or one register transfer (mode other than PWM input waveform measurement mode) can be selected for each channel.
  - Channels for RSPI and SCI: Even channels transfer receive data from a register to the alias area. Odd channels transfer transmit data from the alias area to a register.
  - Channels for RCAN: A channel for reception transfers data in the mailbox to alias area. A channel for transmission transfers data in the alias area to the mailbox.
  - Channels for FlexRay: Contents of a buffer set up as the receive buffer/receive FIFO in the message RAM are transferred to the alias area. Both the header section and data section are transferred to the alias area.
- Specifiable maximum transfer count
  - Channels for ADC and ATU-III (Timer G): 1,023; with a reloading function.
  - Channels for ATU-III (Timer A, C, and F): Transfer count cannot be specified. If transfer is enabled, data transfer is performed whenever a transfer request is accepted without being restricted by the specified transfer count.
  - Channels for RSPI and SCI: 1,023
  - Channel for RCAN (reception): Transfer count cannot be specified. Data in the mailbox is transferred whenever a transfer request is accepted without the restriction of the transfer count in the transfer enable state.
  - Channels for RCAN (transmission): Transmission between up to (A-B-Cch) 93 mailboxes, (A-Bch) 62 mailboxes + (C-Dch) 62 mailboxes can be specified simultaneously.
  - Channels for FlexRay: Transfer count (number of unit transfers) cannot be specified. While transfer on these channels is enabled, data in the buffer are transferred whenever a transfer request is accepted without a restriction on the transfer count.
- Interrupt request: Interrupt requests can be masked for each channel.
  - Channels for ATU-III (Timer G) and ADC: Generate no interrupts.
  - Channels for ATU-III (Timers A, C, and F): Can generate an interrupt request to the CPU after a transfer has been completed.
  - Channels for RSPI and SCI: Can generate an interrupt request to the CPU after the specified count of transfers has been completed.

- Channels for RCAN: A channel for reception does not generate interrupts. A channel for transmission can generate an interrupt request to the CPU after the transmission of the specified mailbox has been completed.
- Channels for FlexRay: No interrupts are generated.

- Address modes
  - Both transfer source and destination addresses are accessed (dual address mode).
- Selectable bus modes
  - Cycle steal mode
- Channel priority: The order of channel priority is fixed (channel 0 > channel 1 > … > channel 74> channel 75…> channel 77).
- Reloading functions: Supported in channels for ATU-III (timer G) and ADC.

The A-DMAC functions are summarized in table 12.1.

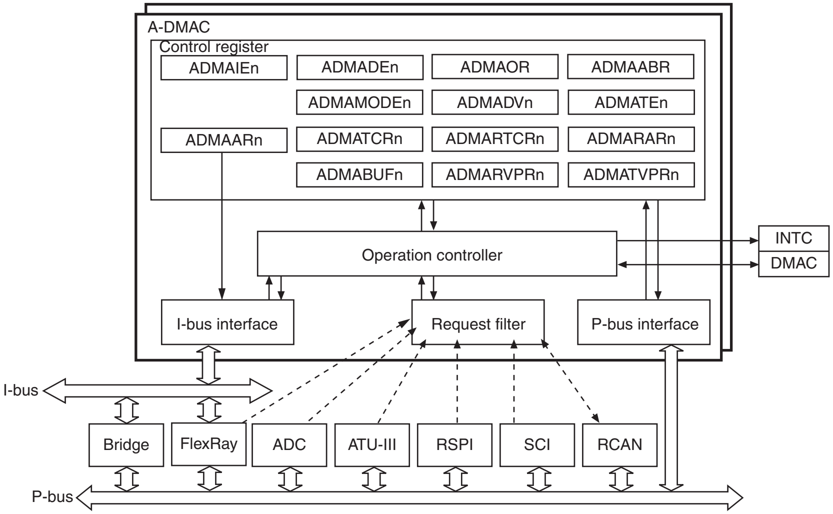
Table 12.1 A-DMAC Channel Functions

	Channels for ATU-III (Timer G)	Channels for ADC	Channels for ATU-III (Timers A, C, and F)	Channels for RSPI	Channels for SCI	Channels for RCAN	Channels for FlexRay
Channel name	0	1	2 to 55	56 to 61	62 to 71	72 to 75	76, 77
Number of channels	1	1	54	6	10	4	2
Transfer request	Timer G2	End of ADC_A and AN0	Ch 2 to 7: Timer A Ch 8 to 27: Timer C Ch 28 to 55: Timer F	Ch 56 & 57: RSPI_A Ch 58 & 59: RSPI_B, Ch 60 & 61: RSPI_C	Ch 62 &63: SCI_A Ch 64 & 65: SCI_B Ch 66 & 67: SCI_C Ch 68 & 69: SCI_D Ch 70 & 71: SCI_E	Ch 72: RCAN_A → RCAN_C Ch 73: RCAN_C → RCAN_D Ch 74 & 75: Software trigger	Ch 76 & 77: FlexRay
Transfer direction	RAM → Port	Register → RAM	Register → RAM	Even channels: Register → RAM Odd channels: RAM → Register	Even channels: Register → RAM Odd channels: RAM → Register	Ch 72 & 73: Register → RAM Ch 74 & 75: RAM → Register	Ch 76 & 77: Register → RAM
Data transfer length	8 bits	16 bits	Ch 2 to 7: 32 bits × 2 Ch 8 to 27: 32 bits Ch 28 to 55: 32 bits × 2 or 32 bits × 1	16 bits	8 bits	20 bytes	Decided by the payload length
Maximum transfer count	1,023 (with a reloading function)	1,023 (with a reloading function)	No restriction	1,023	1,023	No restriction	No restriction

	Channels for ATU-III (Timer G)	**Channels for ADC**	**Channels for ATU-III (Timers A, C, and F)**	**Channels for RSPI**	**Channels for SCI**	**Channels for RCAN**	**Channels for FlexRay**
Interrupt request	No interrupt requests generated	No interrupt requests generated	An interrupt request generated at the end of first transfer	An interrupt request generated at the end of transfer (TCR = 0)	An interrupt request generated at the end of transfer (TCR = 0)	Ch 72 & 73: No interrupt requests generated Ch 74 & 75: An interrupt request generated at the end of transfer	Ch 76 & 77: No interrupt requests generated
Addressing	Transfer source address: Incremented, ring-buffer type transfer*, Transfer destination address: Fixed	Transfer source address: Fixed, Transfer destination address: Incremented, ring-buffer type transfer*	Both transfer source and destination addresses: Fixed	Register address: Fixed, RAM area address: Incremented by two for a transfer	Register address: Fixed, RAM area address: Incremented by two for a transfer	Both transfer source and destination addresses: Incremented	Transfer source address: fixed; transfer destination address: incremented (receive), fixed (receive FIFO)

Note: When TCR = 0, the TCR and alias pointer are reloaded.

Figure 12.1 shows a block diagram of the A-DMAC.



[Legend]

ADMAOR:	A-DMAC operation register	ADMARARn:	A-DMAC reload alias pointer registers (n = 0, 1)
ADMAABR:	A-DMAC alias base register	ADMABUFn:	A-DMAC buffer registers (n = 2 to 7)
ADMAIEn:	A-DMAC interrupt control registers (n = 0 to 9)	ADMARVPRn:	A-DMAC receive wait registers (n = 0 to 7)
ADMAMODEn:	A-DMAC transfer mode registers (n = 0 to 3) (for timer F)	ADMATVPRn:	A-DMAC transmit wait registers (n = 0 to 7)
ADMADVn:	A-DMAC data valid registers (n = 0 to 6)	INTC:	Interrupt controller
ADMATEn:	A-DMAC transfer end registers (n = 0 to 2)	DMAC:	Direct memory access controller
ADMADEn:	A-DMAC enable registers (n = 0 to 7)	ATU-III:	Advanced timer unit III
ADMATCRn:	A-DMAC transfer count registers (n = 0, 1, 56 to 71)	RSPI:	Renesas serial peripheral interface
ADMAARn:	A-DMAC alias pointer registers (n = 0, 1, 56 to 71)	SCI:	Serial communications interface
ADMARTCRn:	A-DMAC reload transfer count registers (n = 0, 1)	RCAN:	Controller area network
		ADC:	A/D converter

Figure 12.1 Block Diagram of A-DMAC

12.1.1 Input/Output Pins

No input/output pins are provided for this module.

12.2 Register Descriptions

The A-DMAC has the registers listed in table 12.2.

Table 12.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Registers common to all channels	A-DMAC operation register	ADMAOR	R/W	H'00	H'FFFE6000	8
	A-DMAC alias base register	ADMAABR	R/W	H'00	H'FFFE6002	8
Registers for each channel	A-DMAC interrupt control register 0	ADMAIE0	R/W	H'00	H'FFFE6010	8
	A-DMAC interrupt control register 1	ADMAIE1	R/W	H'00	H'FFFE6011	8
	A-DMAC interrupt control register 2	ADMAIE2	R/W	H'00	H'FFFE6012	8
	A-DMAC interrupt control register 3	ADMAIE3	R/W	H'00	H'FFFE6013	8
	A-DMAC interrupt control register 4	ADMAIE4	R/W	H'00	H'FFFE6014	8
	A-DMAC interrupt control register 5	ADMAIE5	R/W	H'00	H'FFFE6015	8
	A-DMAC interrupt control register 6	ADMAIE6	R/W	H'00	H'FFFE6016	8
	A-DMAC interrupt control register 7	ADMAIE7	R/W	H'00	H'FFFE6017	8
	A-DMAC interrupt control register 8	ADMAIE8	R/W	H'00	H'FFFE6018	8
	A-DMAC interrupt control register 9	ADMAIE9	R/W	H'00	H'FFFE6019	8
	A-DMAC data valid register 0	ADMADV0	R/(W)*1	H'00	H'FFFE6020	8
	A-DMAC data valid register 1	ADMADV1	R/(W)*1	H'00	H'FFFE6021	8
	A-DMAC data valid register 2	ADMADV2	R/(W)*1	H'00	H'FFFE6022	8
	A-DMAC data valid register 3	ADMADV3	R/(W)*1	H'00	H'FFFE6023	8

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Registers for each channel	A-DMAC data valid register 4	ADMADV4	R/(W)*[1]	H'00	H'FFFE6024	8
	A-DMAC data valid register 5	ADMADV5	R/(W)*[1]	H'00	H'FFFE6025	8
	A-DMAC data valid register 6	ADMADV6	R/(W)*[1]	H'00	H'FFFE6026	8
	A-DMAC transfer end register 0	ADMATE0	R/(W)*[1]	H'00	H'FFFE6030	8
	A-DMAC transfer end register 1	ADMATE1	R/(W)*[1]	H'00	H'FFFE6031	8
	A-DMAC transfer end register 2	ADMATE2	R/(W)*[1]	H'00	H'FFFE6032	8
	A-DMAC enable register 0	ADMADE0	R/W	H'00	H'FFFE6040	8
	A-DMAC enable register 1	ADMADE1	R/W	H'00	H'FFFE6041	8
	A-DMAC enable register 2	ADMADE2	R/W	H'00	H'FFFE6042	8
	A-DMAC enable register 3	ADMADE3	R/W	H'00	H'FFFE6043	8
	A-DMAC enable register 4	ADMADE4	R/W	H'00	H'FFFE6044	8
	A-DMAC enable register 5	ADMADE5	R/W	H'00	H'FFFE6045	8
	A-DMAC enable register 6	ADMADE6	R/W	H'00	H'FFFE6046	8
	A-DMAC enable register 7	ADMADE7	R/W	H'00	H'FFFE6047	8
	A-DMAC transfer mode register 0	ADMAMODE0	R/W	H'00	H'FFFE6050	8
	A-DMAC transfer mode register 1	ADMAMODE1	R/W	H'00	H'FFFE6051	8
	A-DMAC transfer mode register 2	ADMAMODE2	R/W	H'00	H'FFFE6052	8

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Registers for each channel	A-DMAC transfer mode register 3	ADMAMODE3	R/W	H'00	H'FFFE6053	8
	A-DMAC transfer count register 0	ADMATCR0	R/W	H'0000	H'FFFE6060	16
	A-DMAC reload transfer count register 0	ADMARTCR0	R/W	H'0000	H'FFFE6062	16
	A-DMAC transfer count register 1	ADMATCR1	R/W	H'0000	H'FFFE6064	16
	A-DMAC reload transfer count register 1	ADMARTCR1	R/W	H'0000	H'FFFE6066	16
	A-DMAC transfer count register 56	ADMATCR56	R/W	H'0000	H'FFFE6070	16
	A-DMAC transfer count register 57	ADMATCR57	R/W	H'0000	H'FFFE6072	16
	A-DMAC transfer count register 58	ADMATCR58	R/W	H'0000	H'FFFE6074	16
	A-DMAC transfer count register 59	ADMATCR59	R/W	H'0000	H'FFFE6076	16
	A-DMAC transfer count register 60	ADMATCR60	R/W	H'0000	H'FFFE6078	16
	A-DMAC transfer count register 61	ADMATCR61	R/W	H'0000	H'FFFE607A	16
	A-DMAC transfer count register 62	ADMATCR62	R/W	H'0000	H'FFFE607C	16
	A-DMAC transfer count register 63	ADMATCR63	R/W	H'0000	H'FFFE607E	16
	A-DMAC transfer count register 64	ADMATCR64	R/W	H'0000	H'FFFE6080	16
	A-DMAC transfer count register 65	ADMATCR65	R/W	H'0000	H'FFFE6082	16
	A-DMAC transfer count register 66	ADMATCR66	R/W	H'0000	H'FFFE6084	16

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Registers for each channel	A-DMAC transfer count register 67	ADMATCR67	R/W	H'0000	H'FFFE6086	16
	A-DMAC transfer count register 68	ADMATCR68	R/W	H'0000	H'FFFE6088	16
	A-DMAC transfer count register 69	ADMATCR69	R/W	H'0000	H'FFFE608A	16
	A-DMAC transfer count register 70	ADMATCR70	R/W	H'0000	H'FFFE608C	16
	A-DMAC transfer count register 71	ADMATCR71	R/W	H'0000	H'FFFE608E	16
	A-DMAC alias pointer register 0	ADMAAR0	R/W	H'0000	H'FFFE6090	16
	A-DMAC reload alias pointer register 0	ADMARAR0	R/W	H'0000	H'FFFE6092	16
	A-DMAC alias pointer register 1	ADMAAR1	R/W	H'0000	H'FFFE6094	16
	A-DMAC reload alias pointer register 1	ADMARAR1	R/W	H'0000	H'FFFE6096	16
	A-DMAC alias pointer register 56	ADMAAR56	R/W	H'0000	H'FFFE60A0	16
	A-DMAC alias pointer register 57	ADMAAR57	R/W	H'0000	H'FFFE60A2	16
	A-DMAC alias pointer register 58	ADMAAR58	R/W	H'0000	H'FFFE60A4	16
	A-DMAC alias pointer register 59	ADMAAR59	R/W	H'0000	H'FFFE60A6	16
	A-DMAC alias pointer register 60	ADMAAR60	R/W	H'0000	H'FFFE60A8	16
	A-DMAC alias pointer register 61	ADMAAR61	R/W	H'0000	H'FFFE60AA	16
	A-DMAC alias pointer register 62	ADMAAR62	R/W	H'0000	H'FFFE60AC	16
	A-DMAC alias pointer register 63	ADMAAR63	R/W	H'0000	H'FFFE60AE	16
	A-DMAC alias pointer register 64	ADMAAR64	R/W	H'0000	H'FFFE60B0	16

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Registers for each channel	A-DMAC alias pointer register 65	ADMAAR65	R/W	H'0000	H'FFFE60B2	16
	A-DMAC alias pointer register 66	ADMAAR66	R/W	H'0000	H'FFFE60B4	16
	A-DMAC alias pointer register 67	ADMAAR67	R/W	H'0000	H'FFFE60B6	16
	A-DMAC alias pointer register 68	ADMAAR68	R/W	H'0000	H'FFFE60B8	16
	A-DMAC alias pointer register 69	ADMAAR69	R/W	H'0000	H'FFFE60BA	16
	A-DMAC alias pointer register 70	ADMAAR70	R/W	H'0000	H'FFFE60BC	16
	A-DMAC alias pointer register 71	ADMAAR71	R/W	H'0000	H'FFFE60BE	16
	A-DMAC buffer register 2	ADMABUF2	R	H'0000000	H'FFFE60C0	32
	A-DMAC buffer register 3	ADMABUF3	R	H'0000000	H'FFFE60C4	32
	A-DMAC buffer register 4	ADMABUF4	R	H'0000000	H'FFFE60C8	32
	A-DMAC buffer register 5	ADMABUF5	R	H'0000000	H'FFFE60CC	32
	A-DMAC buffer register 6	ADMABUF6	R	H'0000000	H'FFFE60D0	32
	A-DMAC buffer register 7	ADMABUF7	R	H'0000000	H'FFFE60D4	32
	A-DMAC receive wait register 0	ADMARVPR0	R/(W)*2	H'0000	H'FFFE60E0	8, 16
	A-DMAC receive wait register 1	ADMARVPR1	R/(W)*2	H'0000	H'FFFE60E2	8, 16
	A-DMAC receive wait register 2	ADMARVPR2	R/(W)*2	H'0000	H'FFFE60E4	8, 16
	A-DMAC receive wait register 3	ADMARVPR3	R/(W)*2	H'0000	H'FFFE60E6	8, 16
	A-DMAC receive wait register 4	ADMARVPR4	R/(W)*2	H'0000	H'FFFE60E8	8, 16

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Registers for each channel	A-DMAC receive wait register 5	ADMARVPR5	R/(W)*2	H'0000	H'FFFE60EA	8, 16
	A-DMAC receive wait register 6	ADMARVPR6	R/(W)*2	H'0000	H'FFFE60EC	8, 16
	A-DMAC receive wait register 7	ADMARVPR7	R/(W)*2	H'0000	H'FFFE60EE	8, 16
	A-DMAC transmit wait register 0	ADMATVPR0	R/W	H'0000	H'FFFE60F0	8, 16
	A-DMAC transmit wait register 1	ADMATVPR1	R/W	H'0000	H'FFFE60F2	8, 16
	A-DMAC transmit wait register 2	ADMATVPR2	R/W	H'0000	H'FFFE60F4	8, 16
	A-DMAC transmit wait register 3	ADMATVPR3	R/W	H'0000	H'FFFE60F6	8, 16
	A-DMAC transmit wait register 4	ADMATVPR4	R/W	H'0000	H'FFFE60F8	8, 16
	A-DMAC transmit wait register 5	ADMATVPR5	R/W	H'0000	H'FFFE60FA	8, 16
	A-DMAC transmit wait register 6	ADMATVPR6	R/W	H'0000	H'FFFE60FC	8, 16
	A-DMAC transmit wait register 7	ADMATVPR7	R/W	H'0000	H'FFFE60FE	8, 16
	A-DMAC FlexRay receive wait register 0	ADMATVPR0	R/(W)*2	H'0000	H'FFFC2800	8, 16, 32
	A-DMAC FlexRay receive wait register 1	ADMATVPR1	R/(W)*2	H'0000	H'FFFC2804	8, 16, 32
	A-DMAC FlexRay receive wait register 2	ADMATVPR2	R/(W)*2	H'0000	H'FFFC2808	8, 16, 32
	A-DMAC FlexRay receive wait register 3	ADMATVPR3	R/(W)*2	H'0000	H'FFFC280C	8, 16, 32
	A-DMAC FlexRay last message buffer register	ADMAFRLMB	R/W	H'0000	H'FFFC2810	8, 16, 32
	A-DMAC FlexRay general control register	ADMAFRGEN CTR	R/W	H'0000	H'FFFC2811	8
	A-DMAC FlexRay control register	ADMAFRCTR	R/W	H'0000	H'FFFC2812	8, 16

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Registers for each channel	A-DMAC FlexRay FIFO control register	ADMAFRFCTR	R/W	H'0000	H'FFFC2813	8
	A-DMAC FlexRay transmit status register	ADMAFRTRSTAT	R	H'0000	H'FFFC2814	8, 16, 32
	A-DMAC FlexRay status register	ADMAFRSTAT	R	H'0000	H'FFFC2816	8, 16
	A-DMAC FlexRay FIFO status register	ADMAFRFTSTAT	R/(W)*[2]	H'0000	H'FFFC2817	8

Notes: 1. Writing a 0 after reading a 1 is only allowed to clear the flag.

2. Writing a 1 is only allowed to clear the flag.

12.2.1 A-DMAC Operation Register (ADMAOR)

ADMAOR is an 8-bit readable/writable register that specifies operation for all channels.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	DME
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	DME	0	R/W	DMA master enable flag Enables or disables DMA transfer for all channels. If this bit and the DE bit in each channel are set to 1, DMA transfer is enabled. If this bit is cleared to 0, DMA transfer is cancelled for all channels.

Note: If an NMI occurs, the A-DMAC enters the DMA-transfer-disabled state (for details, refer to section 12.3.3, Transfer Suspension and Resumption). To restart DMA transfer, use the NMIF bit in the DMA operation flag bit register (DMAFR) in the DMAC because the A-DMAC does not support such a bit in the ADMAOR register (for details, refer to section 11, Direct Memory Access Controller (DMAC)). Clearing the NMIF bit in the DMAC enables DMA transfer in the A-DMAC.

12.2.2 A-DMAC Alias Base Register (ADMAABR)

ADMAABR is an 8-bit readable/writable register that specifies the start address of the alias area (on-chip RAM are used for transfer for on-chip modules) for the A-DMAC. The alias area can be specified in 32-Kbyte units.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	AA[2:0]		
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	AA[2:0]	000	R/W	Alias area address These bits specify the start address of the alias area. 000: H'FFF80000 001: H'FFF88000 010: H'FFF90000 011: H'FFF98000 100: H'FFFA0000 101: H'FFFA8000 110: H'FFFB0000 111: H'FFFB8000

12.2.3 A-DMAC Interrupt Control Registers (ADMAIE)

ADMAIE is an 8-bit readable/writable register.

Each bit (IE bit) in each ADMAIE register enables or disables an interrupt to the CPU in each channel. If the corresponding DV bit or TE bit is set while the IE bit is set to 1, a transfer end interrupt is requested (for details, refer to section 12.3.6, Transfer Enable/Disable Conditions and Interrupt Requests).

Table 12.3 shows the correspondence between channels and bits.

Bit:	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 12.3 Correspondence between Channels and ADMAIE Registers

Register Name	Bit							
	7	6	5	4	3	2	1	0
ADMAIE0	Channel 7	Channel 6	Channel 5	Channel 4	Channel 3	Channel 2	Reserved	Reserved
ADMAIE1	Channel 15	Channel 14	Channel 13	Channel 12	Channel 11	Channel 10	Channel 9	Channel 8
ADMAIE2	Channel 23	Channel 22	Channel 21	Channel 20	Channel 19	Channel 18	Channel 17	Channel 16
ADMAIE3	Channel 31	Channel 30	Channel 29	Channel 28	Channel 27	Channel 26	Channel 25	Channel 24
ADMAIE4	Channel 39	Channel 38	Channel 37	Channel 36	Channel 35	Channel 34	Channel 33	Channel 32
ADMAIE5	Channel 47	Channel 46	Channel 45	Channel 44	Channel 43	Channel 42	Channel 41	Channel 40
ADMAIE6	Channel 55	Channel 54	Channel 53	Channel 52	Channel 51	Channel 50	Channel 49	Channel 48
ADMAIE7	Channel 63	Channel 62	Channel 61	Channel 60	Channel 59	Channel 58	Channel 57	Channel 56
ADMAIE8	Channel 71	Channel 70	Channel 69	Channel 68	Channel 67	Channel 66	Channel 65	Channel 64
ADMAIE9	Reserved				Channel 75	Channel 74	Reserved	Reserved

Note: A reserved bit is always read as 0. The write value should always be 0.

12.2.4 A-DMAC Data Valid Registers (ADMADV)

ADMADV is an 8-bit readable/writable register.

Each bit (DV bit) in each ADMADV register indicates the DMA transfer state of the corresponding channel. The DV bit is set to 1 to indicate that the data in the alias area is valid when a DMA transfer is completed after the DE bit is set.

Writing 1 to the DV bit is invalid. To clear the DV bit, read 1 from the DV bit and then write 0.

Table 12.4 shows the correspondence between channels and bits.

Bit:	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * To clear the flag, write 0 to the bit to be cleared, only when 1 has been read from the bit when these registers have been read. To the bit not to be cleared, only 1 can be written, even if 0 is read from the bit. To the reserved bits, only 0 can be written. When the flag is read by the CPU, even though 0 has been read, the flag may be set to 1 and therefore 1 may be read internally. In this case, if 0 is written to the corresponding bit, the flag will be cleared even if 1 has not been read by the CPU. Since the specifications of the flag bits in these registers differ from the ones of the flags in other modules, the clearing operations of the flags by the CPU must be executed carefully.

Table 12.4 Correspondence between Channels and ADMADV Registers

Register	Bit							
Name	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ADMADV0	Channel 7	Channel 6	Channel 5	Channel 4	Channel 3	Channel 2	Reserved	Reserved
ADMADV1	Channel 15	Channel 14	Channel 13	Channel 12	Channel 11	Channel 10	Channel 9	Channel 8
ADMADV2	Channel 23	Channel 22	Channel 21	Channel 20	Channel 19	Channel 18	Channel 17	Channel 16
ADMADV3	Channel 31	Channel 30	Channel 29	Channel 28	Channel 27	Channel 26	Channel 25	Channel 24
ADMADV4	Channel 39	Channel 38	Channel 37	Channel 36	Channel 35	Channel 34	Channel 33	Channel 32
ADMADV5	Channel 47	Channel 46	Channel 45	Channel 44	Channel 43	Channel 42	Channel 41	Channel 40
ADMADV6	Channel 55	Channel 54	Channel 53	Channel 52	Channel 51	Channel 50	Channel 49	Channel 48

Note: A reserved bit is always read as 0. The write value should always be 0.

12.2.5 A-DMAC Transfer End Registers (ADMATE)

ADMATE is an 8-bit readable/writable register.

Each bit (TE bit) in each ADMATE register indicates the DMA transfer state of the corresponding channel. The TE bit is set to 1 when the specified counts of DMA transfers are completed.

Writing 1 to the TE bit is invalid. To clear the TE bit, read 1 from the TE bit and then write 0 to it. While the TE bit is set to 1, the corresponding channel cannot accept the transfer request.

Table 12.5 shows the correspondence between channels and ADMATE registers.

Bit:	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * To clear the flag, write 0 to the bit to be cleared, only when 1 has been read from the bit when these registers have been read. To the bit not to be cleared, only 1 can be written, even if 0 is read from the bit. To the reserved bits, only 0 can be written. When the flag is read by the CPU, even though 0 has been read, the flag may be set to 1 and therefore 1 may be read internally. In this case, if 0 is written to the corresponding bit, the flag will be cleared even if 1 has not been read by the CPU. Since the specifications of the flag bits in these registers differ from the ones of the flags in other modules, the clearing operations of the flags by the CPU must be executed carefully.

Table 12.5 Correspondence between Channels and ADMATE Registers

Register Name	Bit							
	7	6	5	4	3	2	1	0
ADMATE0	Channel 63	Channel 62	Channel 61	Channel 60	Channel 59	Channel 58	Channel 57	Channel 56
ADMATE1	Channel 71	Channel 70	Channel 69	Channel 68	Channel 67	Channel 66	Channel 65	Channel 64
ADMATE2	Reserved				Channel 75	Channel 74	Reserved	Reserved

Note: A reserved bit is always read as 0. The write value should always be 0.

12.2.6 A-DMAC Enable Registers (ADMADE)

ADMADE is an 8-bit readable/writable register.

Each bit (DE bit) in each ADMADE register enables or disables DMA transfer for a channel for ATU-III (timers A, C, and F), RCAN, or FlexRay. Setting the DE bit of the corresponding channel and the DME bit in ADMAOR to 1 enables DMA transfer.

Table 12.6 shows the correspondence between channels and bits.

Bit:	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 12.6 Correspondence between Channels and ADMADE Registers

Register Name	Bit							
	7	6	5	4	3	2	1	0
ADMADE0	Channel 7	Channel 6	Channel 5	Channel 4	Channel 3	Channel 2	Reserved	Reserved
ADMADE1	Channel 15	Channel 14	Channel 13	Channel 12	Channel 11	Channel 10	Channel 9	Channel 8
ADMADE2	Channel 23	Channel 22	Channel 21	Channel 20	Channel 19	Channel 18	Channel 17	Channel 16
ADMADE3	Channel 31	Channel 30	Channel 29	Channel 28	Channel 27	Channel 26	Channel 25	Channel 24
ADMADE4	Channel 39	Channel 38	Channel 37	Channel 36	Channel 35	Channel 34	Channel 33	Channel 32
ADMADE5	Channel 47	Channel 46	Channel 45	Channel 44	Channel 43	Channel 42	Channel 41	Channel 40
ADMADE6	Channel 55	Channel 54	Channel 53	Channel 52	Channel 51	Channel 50	Channel 49	Channel 48
ADMADE7	Reserved	Reserved	Channel 77*[1]	Channel 76*[1]	Channel 75	Channel 74	Channel 73*[2]	Channel 72*[2]

Notes: A reserved bit is always read as 0. The write value should always be 0.

1. When disabling transmission on a channel for FlexRay after having enabling it, do so in accord with the procedure given in section 12.3.2 (6), Suspension and resumption of transfer on a FlexRay channel.
2. Transfer channel for RCAN is set in ADMADE7[1:0] as shown below. Setting of a channel for reception should be made before the settings for RCAN operation. If the setting are changed during RCAN operation, operations for the transfer of received data to the alias area are not guaranteed. For operations when the settings are changed to the transfer disabled condition during the transfer operation, refer to section 12.3.3, Transfer Suspension and Resumption.

Transfer channel setting	ADMADE7[1]	ADMADE7[0]
Transfer diasbled	0	0
RCAN_A-RCAN_C	0	1
RCAN_C-RCAN_D	1	0
RCAN_A-RCAN_D	1	1

12.2.7 A-DMAC Transfer Mode Registers (ADMAMODE)

ADMAMODE is an 8-bit readable/writable register.

Each bit (MODE bit) in each ADMAMODE register specifies the transfer mode for each channel used for ATM-III (timer F). A write to this register should be performed in the transfer-disabled state (DE = 0 or DME = 0).

To use timer F in PWM input waveform measurement mode, set the MODE bit to 1; otherwise, clear the MODE bit to 0.

Table 12.7 shows the correspondence between channels and the ADMAMODE register.

Bit:	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 12.7 Correspondence between Channels and ADMAMODE Registers

Register Name	Bit							
	7	6	5	4	3	2	1	0
ADMAMODE0	Channel 31	Channel 30	Channel 29	Channel 28	Reserved	Reserved	Reserved	Reserved
ADMAMODE1	Channel 39	Channel 38	Channel 37	Channel 36	Channel 35	Channel 4	Channel 33	Channel 32
ADMAMODE2	Channel 47	Channel 46	Channel 45	Channel 44	Channel 43	Channel 42	Channel 41	Channel 40
ADMAMODE3	Channel 55	Channel 54	Channel 53	Channel 52	Channel 51	Channel 50	Channel 49	Channel 48

Note: A reserved bit is always read as 0. The write value should always be 0.

12.2.8 A-DMAC Transfer Count Registers (ADMATCR)

ADMATCR is a 16-bit readable/writable register that specifies the DMA transfer count for channels used for ATU-III (timer G), ADC, RSPI, and SCI.

While DMA transfer is enabled, the ADMATCR value is updated (post-decremented) each time a DMA transfer is performed, thus indicating the remaining transfer count till transfer end. The ADMATCR registers can be specified from H'0000 to H'03FF.

Clearing ADMATCR to 0 suspends the DMA transfer. Note that writing a value other than 0 to ADMATCR while ADMATCR ≠ 0 is prohibited.

The ADMARTCR value is reloaded into ADMATCR corresponding to channels for ATU-III (timer G) and ADC, if a DMA transfer is performed while ADMATCR = 1 and ADMARTCR ≠ 0.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

12.2.9 A-DMAC Alias Pointer Registers (ADMAAR)

ADMAAR is a 16-bit readable/writable register that specifies the alias area for channels used for ATU-III (timer G), ADC, RSPI, and SCI. The ADMAAR value is updated (post-updated) to indicate the next transfer address each time a DMA transfer is performed. ADMAAR must be written while the corresponding ADMATCR is cleared to 0.

The alias area for each channel is defined as an area starting from the address specified by ADMAAR to the address obtained by the transfer count specified by ADMATCR × 2 bytes (for channels used for RSPI and ADC) or to the address obtained by the transfer count specified by ADMATCR × 1 byte (for channels used for the SCI and ATU-III (timer G)).

The transfer destination address is specified by the relative address from the start address of the alias area. The specifiable relative address is from H'0000 to H'7FFF.

The ADMARAR value will be reloaded into ADMAAR corresponding to channels for ATU-III (timer G) and the ADC, if a DMA transfer is performed while ADMATCR = 1 and ADMARTCR ≠ 0.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W*

Note: * A writing to bit 0 in the ADMAAR registers for channels used for ADC and RSPI is invalid.

12.2.10 A-DMAC Reload Transfer Count Registers (ADMARTCR)

ADMARTCR is a 16-bit readable/writable register that specifies reload values of the corresponding ADMATCR in a channel used for ADC. ADMARTCR can be specified from H'0000 to H'03FF.

The ADMARTCR value will be reloaded into ADMATCR if a DMA transfer is performed while ADMATCR = 1 and ADMARTCR ≠ 0.

ADMARTCR must be set before ADMATCR is set.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

12.2.11 A-DMAC Reload Alias Pointer Registers (ADMARAR)

ADMARAR is a 16-bit readable/writable register that specifies reload values of the corresponding ADMAAR in a channel used for ADC. ADMARAR can be specified from H'0000 to H'7FFF.

The ADMARAR value is reloaded into ADMAAR if a DMA transfer is performed while ADMATCR = 1 and ADMARTCR ≠ 0.

ADMARAR must be set before the ADMAAR registers are set.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W*

Note: * A writing to bit 0 in the ADMARAR registers for channels used for ADC is invalid.

12.2.12 A-DMAC Buffer Registers (ADMABUF)

ADMABUF is a 32-bit read-only register. ADMABUF2 to ADMABUF7 correspond to channels 2 to 7 for timer A.

ADMABUF stores the value read for the input capture register according to the transfer request from timer A. Two values: data stored in ADMABUF (value read for the input capture register for the previous transfer request from timer A) and the value read for the input capture register from timer A will be transferred to the alias area using the timer A transfer request as a trigger.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

12.2.13 A-DMAC Receive Wait Registers (ADMARVPR)

ADMARVPR is a 16-bit readable/writable register that includes the flags corresponding to mailboxes (MBx) of the RCAN module. Only 1 can be written to ADMARVPR to clear flags in ADMARVPR.

Table 12.8 shows the correspondence between ADMARVPR and mailboxes in the RCAN.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * To clear the flag, write 1 to the bit to be cleared, only when 1 has been read from the bit when these registers have been read.
To the bit not to be cleared, only 0 can be written, even if 0 is read from the bit. To the reserved bits, only 0 can be written.
When the flag is read by the CPU, even though 0 has been read, the flag may be set to 1 and therefore 1 may be read internally. In this case, if 1 is written to the corresponding bit, the flag will be cleared even if 1 has not been read by the CPU.
Since the specifications of the flag bits in these registers differ from the ones of the flags in other modules, the clearing operations of the flags by the CPU must be executed carefully.

Table 12.8 Correspondence between ADMARVPR and Mailboxes in RCAN

ADMARVPR	Bits	RCAN Module	MBx
ADMARVPR0	15 to 0	RCAN_A	31 to 16
ADMARVPR1	15 to 0		15 to 0
ADMARVPR2	15 to 0	RCAN_B	31 to 16
ADMARVPR3	15 to 0		15 to 0
ADMARVPR4	15 to 0	RCAN_C	31 to 16
ADMARVPR5	15 to 0		15 to 0
ADMARVPR6	15 to 0	RCAN_D	31 to 16
ADMARVPR7	15 to 0		15 to 0

A flag in ADMARVPR corresponding to the mailbox in the RCAN module is set to 1 when data in the mailbox in the RCAN module has been transferred to the alias area according to the transfer request from the RCAN. To clear a flag (RV bit) in ADMAPVPR, read 1 from the flag (RV bit) and then write 1 to it again.

12.2.14 A-DMAC Transmit Wait Registers (ADMATVPR)

ADMATVPR is a 16-bit readable/writable register that includes the flags corresponding to mailboxes (MBx) of the RCAN module.

ADMATVPR specifies the IDs of mailboxes, to which data is transferred from the alias area, by software. If a bit corresponding to the mailbox (MBx) is set to 1, data is transferred from the alias area to the mailbox (MBx) in transfer enable state (DME = 1, DE = 1, and TE = 0).

When data transfer is completed, the flag corresponding to the mailbox (MBx) is cleared to 0 by hardware.

A bit corresponding to a mailbox (MBx), which is not specified as a transmitter mailbox in the RCAN modules, cannot be set to 1. Note that all the ADMATVPR registers cannot be written to while the RCAN transmitter channel is in transfer enable state or in suspend state (for details, refer to section 12.3.3, Transfer Suspension and Resumption).

Table 12.9 shows the correspondence between ADMATVPR registers and mailboxes in RCAN modules.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 12.9 Correspondence between ADMATVPR Registers and Mailboxes in RCAN Modules

ADMARVPR	Bits	RCAN Module	MBx
ADMATVPR0	15 to 0	RCAN_A	31 to 16
ADMATVPR1	15 to 1*		15 to 1*
ADMATVPR2	15 to 0	RCAN_B	31 to 16
ADMATVPR3	15 to 1*		15 to 1*
ADMATVPR4	15 to 0	RCAN_C	31 to 16
ADMATVPR5	15 to 1*		15 to 1*
ADMATVPR6	15 to 0	RCAN_D	31 to 16
ADMATVPR7	15 to 1*		15 to 1*

Note: * MB0 cannot be specified because the MB0 is a mailbox for reception. Accordingly, a write to bit 0 of the corresponding ADMATVTR register is invalid.

12.2.15 A-DMAC FlexRay Receive Wait Registers (ADMAFRWR)

ADMAFRWR is a 32-bit readable/writable register made up of flags corresponding to message buffer channels. The only writable value for any bit of ADMAFRWR is 1, which clears the corresponding flag.

Table 12.10 shows the correspondence between ADMAFRWR and Mailboxes in FlexRay.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Table 12.10 Correspondence between ADMAFRWR and Mailboxes in FlexRay

ADMAFRWR	Bits	FlexRay Module	MBx
ADMAFRWR3	31 to 0	RWF127 to RWF96	127 to 96*
ADMAFRWR2	31 to 0	RWF95 to RWF64	95 to 64*
ADMAFRWR1	31 to 0	RWF63 to RWF32	63 to 32*
ADMAFRWR0	31 to 0	RWF31 to RWF0	31 to 0*

Note: The set of enabled message buffers will vary with the setting of the ADLNG bits in the ADMAFRCTR register.

In ADMAFRWR, when the contents of the message buffer channel configured as receive buffer according to the transfer request from FlexRay to the alias areas are transferred, the flag of RWF corresponding to the message buffer channel is set to 1. When the RWF flag is set to 1, even the received data is stored to the corresponding message buffer channel, a transfer to the alias areas is not performed. To clear the RWF to 0, Read 1 of RWF flag first and write 1. Receive data transfer mode and the set condition of RWF flag at A-DMAC vary with the setting of ADPEHM bit in A-DMAC FlexRay general control register (ADMAFRGENCTR).

Table 12.11 shows the data transfer mode and RWF flag mode when a parity error occurred.

Table 12.11 The Data Transfer Mode and RWF Flag Mode When a Parity Error Occurred

			No parity error	Detection of an error in the message buffer	Occurrence of a parity error in the RAM1 or 2 output buffer
ADPEHM bit setting	0	Data transfer by A-DMAC	Transfer proceeds	Transfer operations are stopped. *2	Transfer operations are stopped. *2
		State of RWF	Set to 1 after transfer*1	Set the value other than 1*3	Set the value other than 1*3
	1	Data transfer by A-DMAC	Transfer proceeds	Transfer proceeds	Transfer proceeds
		State of RWF	Set to 1 after transfer*4	Set to 1 after transfer*4	Set to 1 after transfer*4

Notes: 1. This is the case when there is no parity error in the data in the user RAM for which the RWF flag has been set.

2. When a parity error has occurred, the A-DMAC does not transfer data to user RAM even though data have been received in the message RAM for FlexRay. If the transfer is to be restarted, clear the faulty-message-buffer-detected flag (FMBD) and the output RAM1, 2 parity error flag (POBF) in the FlexRay message handler status register (FRMHDS) of the FlexRay module, write 0 to the ADENC bit in the A-DMAC FlexRay control register (ADMAFRCTR), and then write 1 to the same bit. Refer to section 12.3.2 (7), Operation when a parity error occurs in a FlexRay channel.

3. Within the FlexRay module, the new data flag in the FlexRay new data register n (FRNDATn) for the channel corresponding to the parity error is cleared.

4. Received data that have been transferred to the user RAM may have a parity error. Use software to implement measures to check the validity of the data by, for example, using the parity-error interrupt of the FlexRay module.

12.2.16 A-DMAC FlexRay Last Message Buffer Registers (ADMAFRLMB)

ADMAFRLMB is an 8-bit readable/writable register for setting the last channel number of the receive setting message buffer of the FlexRay module to have been transferred by the A-DMAC. The value to be set in this register is less than the value of the FIFO buffer start address position setting bits (FFB0 to FFB7) in the FlexRay message RAM setting register (FRMRC) of the FlexRay module, and less than or equal to the value of the last buffer number setting bits (LCB0 to LCB7). Also, the message-buffer numbers which can be set in the LADMB bits depend on the value of the A-DMAC data handling length bits (ADLNG) in the A-DMAC FlexRay control register (ADMAFRCTR) in the way shown in table 12.12.

Bit:	7	6	5	4	3	2	1	0
	-	LAD MB6	LAD MB5	LAD MB4	LAD MB3	LAD MB2	LAD MB1	LAD MB0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6 to 0	LADMB[6:0]	All 0	R/W	A-DMAC specifies the last channel number for a message buffer of the FlexRay module which has been transferred to the alias area.

Table 12.12 Setting value for the LADMB register

ADLNG[1:0]	Target channel number for the transfer	Channel number which can be set in the LADMB
00	0 to 127	0 to 127
01	0 to 75	0 to 75
10	0 to 41	0 to 41
11	0 to 21	0 to 21

12.2.17 A-DMAC FlexRay General Control Register (ADMAFRGENCTR)

ADMAFRGENCTR is an 8-bit readable/writable register that holds a setting for control of A-DMAC transfer operation in cases where a parity error has been detected when received data or data received in the FIFO are transferred from the message RAM for the FlexRay module to user RAM by using the A-DMAC.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	ADP EHM
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	ADPEHM	0	R/W	This bit sets the transfer operation in cases where received data or data received in the FIFO and transferred to the user RAM by using the A-DMAC have a parity error. 0: Transfer operation of the A-DMAC is suspended if there is a parity error. Specifically, data transfer by the A-DMAC is suspended if the output buffer RAM1, 2 parity error flag (POBF) or faulty-message-buffer-detected flag (FMBD) in the FlexRay message handler status register is set to 1. 1: Transfer operation of the A-DMAC is continued even if there is a parity error. Specifically, data transfer by the A-DMAC is not suspended if the output buffer RAM1, 2 parity error flag (POBF) or error message detect flag (FMBD) in the FlexRay message handler status register is set to 1.

Note: The conditions for setting of the individual flags of the A-DMAC FlexRay Receive Wait Register, and the FRWF flag of the A-DMAC FlexRay FIFO status register (ADMAFRFSTAT) are different. Refer to 12.2.21, A-DMAC FlexRay Status Register (ADMAFRSTAT), and section 12.2.22, A-DMAC FlexRay FIFO Status Register (ADMAFRFSTAT).

12.2.18 A-DMAC FlexRay Control Register (ADMAFRCTR)

ADMAFRCTR is an 8-bit readable/writable register that holds the setting for the length of payloads for the message buffer transferred by the channel for the receive message buffer of the FlexRay module and also controls output by the FlexRay module of requests for data transfer from the receive message buffers to the A-DMAC. The setting for payload length is effective for all receive message buffers. In addition, the maximum number of message buffer channels in the FlexRay module that are to be transferred to the alias area varies according to the setting for payload length. The message buffer channels to be transferred to the alias area are shown in table 12.13. After this register has been used to set up the output of transfer requests to the A-DMAC, do not use the CPU to read from FlexRay data section read register n (FRRDDSn). If reading by the CPU does proceed, the detection of parity errors in the transfer of received data by the A-DMAC is not guaranteed.

Bit:	7	6	5	4	3	2	1	0
	ADLNG[1:0]		-	-	-	-	-	ADENC
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7, 6	ADLNG[1:0]	00	R/W	Payload-length setting for receive message buffers in the FlexRay module that are to be transferred by the A-DMAC. 00: Payload length is 32 bytes. 01: Payload length is 64 bytes. 10: Payload length is 128 bytes. 11: Payload length is 254 bytes.
5 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	ADENC	0	R/W	This bit controls whether or not transfer requests are issued to the receive message buffer transfer channel of the FlexRay module. The setting of this bit can change the state of the ADSTAT bit in the A-DMAC FlexRay status register (ADMAFRSTAT).* 0: Output of transfer requests from the receive message buffer to the A-DMAC is disabled. 1: Output of transfer requests from the receive message buffer to the A-DMAC is enabled.

Note: The state of the ADSTAT bit can change even when a parity error occurs.

Table 12.13 Correspondence between payload length and message buffer channels to be transferred

Payload length (bytes)	Message buffer channels in the FlexRay module that are to be transferred to the alias area
32	Message buffer0 to message buffer 127
64	Message buffer0 to message buffer 75
128	Message buffer0 to message buffer 41
254	Message buffer0 to message buffer 21

12.2.19 A-DMAC FlexRay FIFO Control Register (ADMAFRFCTR)

ADMAFRFCTR is an 8-bit readable/writable register that holds the setting for the length of payloads for the message buffer transferred by the channel for the receive FIFO message buffer of the FlexRay module and also controls output by the FlexRay module of requests for data transfer from the receive FIFO message buffers to the A-DMAC.
After this register has been used to set up the output of transfer requests to the A-DMAC, do not use the CPU to read from FlexRay data section read register n (FRRDDSn). If reading by the CPU does proceed, the detection of parity errors in the transfer of received data by the A-DMAC is not guaranteed.

Bit:	7	6	5	4	3	2	1	0
	ADFLNG[1:0]		-	-	-	-	-	ADF ENC
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7, 6	ADFLNG[1:0]	00	R/W	Payload-length setting for receive FIFO message buffers in the FlexRay module that are to be transferred by the A-DMAC. 00: Payload length is 32 bytes. 01: Payload length is 64 bytes. 10: Payload length is 128 bytes. 11: Payload length is 254 bytes.
5 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	ADFENC	0	R/W	This bit controls whether or not transfer requests are issued to the receive FIFO message buffer transfer channel of the FlexRay module. The setting of this bit can change the state of the ADFSTAT bit in the A-DMAC FlexRay FIFO status register (ADMAFRFSTAT).* 0: Output of transfer requests from the receive FIFO message buffer to the A-DMAC is disabled. 1: Output of transfer requests from the receive FIFO message buffer to the A-DMAC is enabled.

Note: The state of the ADFSTAT bit can change even when a parity error occurs.

12.2.20 A-DMAC FlexRay Transfer Status Register (ADMAFRTRSTAT)

ADMAFRSTAT is an 8-bit readable register that indicates the message buffer number of the FlexRay receive message buffer channel or receive FIFO message buffer channel during transfer by the A-DMAC to user RAM from the message RAM of the FlexRay module.

Bit:	7	6	5	4	3	2	1	0
	ADT MB[7]	ADT MB[6]	ADT MB[5]	ADT MB[4]	ADT MB[3]	ADT MB[2]	ADT MB[1]	ADT MB[0]
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	ADTMB[7:0]	All 0	R	During transfer by the A-DMAC of data from a receive message buffer channel or receive FIFO message buffer channel (the message RAM of the FlexRay module) to user RAM, these bits indicate the number of the message buffer.*[1] When transfer by the A-DMAC is not in progress or the setting of the ADPEHM bit of A-DMAC FlexRay general control register (ADMAFRGENCTR) is 0 so that the setting for transfer operation of the A-DMAC in case of a parity error is suspension, this is the number of the last message buffer to have been transferred by the A-DMAC.*[1]

Note: When the target for transfer is a receive FIFO message buffer, these bits indicate the setting for the FIFO buffer start position bits (FFB0 to FFB7) in the FlexRay message RAM setting register (FRMRC) of the FlexRay module.

12.2.21 A-DMAC FlexRay Status Register (ADMAFRSTAT)

ADMAFRSTAT is an 8-bit readable register that indicates whether a parity error occurred or not and a transfer request output condition to A-DMAC of channel for the receive message buffer during a transfer.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	ADDPE	ADSTAT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	ADDPE	0	R	This bit indicates whether or not a parity error has occurred during channel transfer for the receive message buffer. This flag is only enabled when the value of the ADPEHM bit of the A-DMAC FlexRay general control register (ADMAFRGENCTR) is 0. If the ADPEHM bit is set to 1, this flag will not be set to 1 even if there is a parity error. 0: A parity error has not occurred Clearing conditions: • Power on reset • Clearing of the faulty-message-buffer-detected flag (FMBD) or the output RAM1, 2 parity error flag (POBF) in the FlexRay message handler status register (FRMHDS) of the FlexRay module, and writing of 0 to the ADENC bit in the A-DMAC FlexRay control register (ADMAFRCTR) followed by writing of 1 to the same bit. 1: A parity error has occurred Setting condition: In transfer by the A-DMAC of a buffer set up for reception from message RAM in the FlexRay module to user RAM, the faulty-message-buffer-detected flag (FMBD) or the output RAM1, 2 parity error flag (POBF) in the FlexRay message handler status register (FRMHDS) of the FlexRay module is set to 1.

Bit	Bit Name	Initial Value	R/W	Description
0	ADSTAT	0	R	This bit indicates the state of the output of transfer requests from the transfer channels for receive-message buffers. 0: The output of requests for transfer by the A-DMAC of receive message buffers is disabled. Clearing conditions: • Power on reset • Clearing of the ADENC bit in the A-DMAC FlexRay control register (ADMAFRCTR) to 0. • The output buffer RAM1, 2 parity error flag (POBF) or the faulty-message-buffer-detected flag (FMBD) in the FlexRay message handler status register (FRMHDS) being set to 1 during the transfer of a receive buffer by the A-DMAC while the value of the ADPEHM bit in the A-DMAC FlexRay general control register (ADMAFRGENCTR) is 0. 1: The output of requests for transfer by the A-DMAC of receive message buffers is enabled. Setting condition: The ADENC bit of ADMAFRCTR being set to 1.

Note: Refer to section 12.3.2 (7), Operation when a parity error occurs in a FlexRay channel.

12.2.22 A-DMAC FlexRay FIFO Status Registers (ADMAFRFSTAT)

ADMAFRSTAT is an 8-bit readable register that has the corresponding flag to message buffer channel which has been set to receive FIFO in FlexRay. Also, this register indicates a mode of transfer request output to A-DMAC of channels for receive message buffer.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	FRWF	ADF DPE	ADF STAT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/(W)*	R	R

Note: * The only effective write operation is writing 1 to the bit after having read it as 1, which clears the flag.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	FRWF	0	R/(W)*[1]	This flag is set to 1 on completion of the transfer of the contents of a receive FIFO message buffer in message RAM to the alias area in response to a transfer request for a receive FIFO message buffer. While the FRWF flag is set to 1, transfer to the alias area will not proceed even if receive FIFO data is stored in the receive FIFO channel. To clear the FRWF to 0, read it as 1 and then write 1 to it.

Bit	Bit Name	Initial Value	R/W	Description
1	ADFDPE	0	R	This bit indicates whether or not there has been a parity error in channel transfer for a receive FIFO message buffer. This flag is only enabled when the ADPEHM bit of the A-DMAC FlexRay general control register (ADMAFRGENCTR) is 0. When ADPEHM is 1, ADFDPE will not be set to 1 even if a parity error occurs. 0: A parity error has not occurred Clearing conditions: • Power on reset • Clearing of the faulty-message-buffer-detected flag (FMBD) or the output RAM1, 2 parity error flag (POBF) in the FlexRay message handler status register (FRMHDS) of the FlexRay module, and writing of 0 to the ADENC bit in the A-DMAC FlexRay control register (ADMAFRCTR) followed by writing of 1 to the same bit. 1: A parity error has occurred Setting condition: In transfer by the A-DMAC of a buffer set up for FIFO reception from message RAM in the FlexRay module to user RAM, the faulty-message-buffer-detected flag (FMBD) or the output RAM1, 2 parity error flag (POBF) in the FlexRay message handler status register (FRMHDS) of the FlexRay module is set to 1.

Bit	Bit Name	Initial Value	R/W	Description
0	ADFSTAT	0	R	This bit indicates the state of the output of transfer requests from the transfer channels for receive FIFO message buffers of the FlexRay module. 0: The output of requests for transfer by the A-DMAC of receive FIFO message buffers is disabled. Clearing conditions: • Power on reset • Clearing of the ADFENC bit in the A-DMAC FlexRay FIFO control register (ADMAFRFCTR) to 0. • A-DMAC FlexRay general control register: The output buffer RAM1, 2 parity error flag (POBF) or the faulty-message-buffer-detected flag (FMBD) in the FlexRay message handler status register (FRMHDS) being set to 1 during the transfer of a receive FIFO buffer by the A-DMAC while the value of the ADPEHM bit in the A-DMAC FlexRay general control register (ADMAFRGENCTR) is 0 1: The output of requests for transfer by the A-DMAC of receive FIFO message buffers is enabled. Setting conditions: The ADFENC bit of ADMAFRFCTR being set to 1.

Notes: 1. The only effective write operation is writing 1 to the bit after having read it as 1, which clears the flag.

2. Refer to section 12.3.2 (7), Operation when a parity error occurs in a FlexRay channel.

Receive FIFO data transfer status and the setting conditions for FRWF flag vary depending on the settings for ADPEHM bit of A-DMAC FlexRay general control register (ADMAFRGENCTR).

Table 12.14 shows the mode of data transfer at a parity error occurrence and FRWF flag mode.

Table 12.14 Mode of Data Transfer at a Parity Error Occurrence and FRWF Flag Mode

			No parity error	**Detection of an error in the message buffer**	**Occurrence of a parity error in the RAM1, 2 output buffer**
Setting of ADPEHM bit	0	Data transfer by A-DMAC	Transfer proceeds	Transfer operations are stopped.*[2]	Transfer operations are stopped.*[2]
		State of FRWF	Set to 1 after transfer	Set to a value other than 1*[3]	Set to a value other than 1*[3]
	1	Data transfer by A-DMAC	Transfer proceeds	Transfer proceeds	Transfer proceeds
		State of FRWF	Set to 1 after transfer*[4]	Set to 1 after transfer*[4]	Set to 1 after transfer*[4]

Notes: 1. A parity error is not occurring to the data on the user RAM which has been set FRWF flag.

2. When a parity error is occurred, even receive FIFO data is in the message RAM of FlexRay, A-DMAC does not perform a data transfer to the user RAM. Clear the default message buffer detect flag (FMBD), output buffer RAM1, 2 parity error flag (POBF) of FlexRay message handler status register (FRMHDS) first, and write 0 to the ADFENC bit of A-DMAC FlexRay FIFO control register (ADMAFRFCTR), then write 1 again. Refer to section 12.3.2 (7), Operation when a parity error occurs in a FlexRay channel, for details.

3. New data flag in the corresponding channel of the FlexRay new data register n (FRNDATn) in the FlexRay where a parity error occurred is cleared.

4. Receive FIFO data that is transferred to the user RAM may occurring a parity error. As for the effectiveness of data, implement the countermeasures with the software such as using a parity error interrupt of FlexRay.

12.3 Operation

12.3.1 Alias Areas

Areas in on-chip RAM that can be used as a transfer source and destination for A-DMAC data transfer are called alias areas.

The start address of an alias area can be specified by ADMAABR in 32-Kbyte units. Figure 12.2 shows the memory map of the alias areas (ADMAABR is specified as default value).

In channels for ATU-III (timers A, C, and F) and RCAN, and FlexRay, the relative address corresponding to the address specified by ADMAABR for each channel alias area is fixed. For details, refer to section 12.3.2 (2), Operation for A-DMAC Channels Used for ATU-III (timers A, C, and F) and section 12.3.2 (4), Operation for A-DMAC Channels Used for RCAN.

While in channels for ATU-III (timer G), ADC, RSPI, and SCI, the alias area for each channel can be specified. The start address and area size of each alias area are specified by the ADMAAR and ADMATCR registers, respectively. For details, refer to section 12.2.8, A-DMAC Transfer Count Registers (ADMATCR), section 12.2.9, A-DMAC Alias Pointer Registers (ADMAAR), and section 12.3.2 (5), Operation for FlexRay Channels.

Note: Illegal settings such as alias area overlap and alias area setting exceeding the on-chip RAM maximum address can be specified according to the ADMAAR and ADMATCR settings. These illegal settings cannot be detected by hardware.

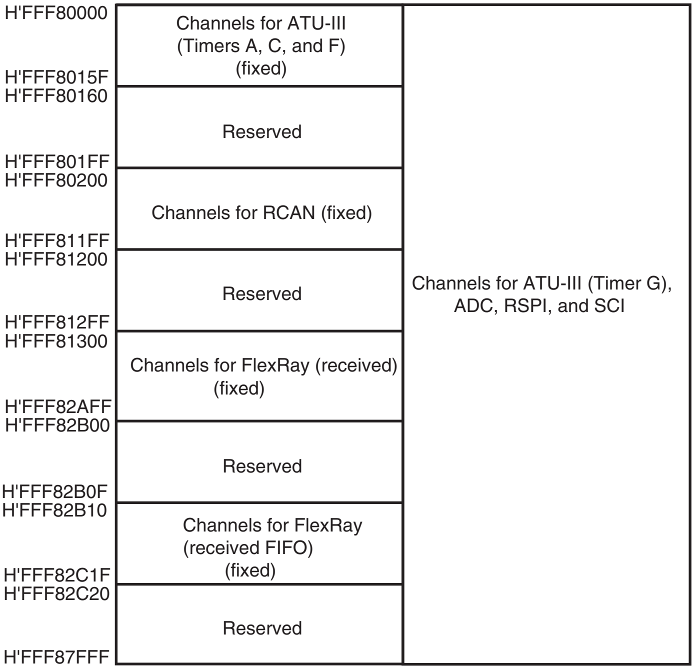

H'FFF80000
H'FFF8015F
H'FFF80160
H'FFF801FF
H'FFF80200
H'FFF811FF
H'FFF81200
H'FFF812FF
H'FFF81300
H'FFF82AFF
H'FFF82B00
H'FFF82B0F
H'FFF82B10
H'FFF82C1F
H'FFF82C20
H'FFF87FFF
Channels for ATU-III (Timers A, C, and F) (fixed)
Reserved
Channels for RCAN (fixed)
Reserved
Channels for FlexRay (received) (fixed)
Reserved
Channels for FlexRay (received FIFO) (fixed)
Reserved
Channels for ATU-III (Timer G), ADC, RSPI, and SCI

Figure 12.2 Memory Map of Alias Areas

12.3.2 Each A-DMAC Channel Operation

(1) Operation for A-DMAC Channels Used for ATU-III (Timer G) and ADC

- Overview

  A-DMAC channel (channel 0) for the ATU-III (timer G) supports the ADC input signal switch setting via a port. This channel performs data transfer from the alias area to port G that controls the ADC input using a transfer request from timer G as a source.

  A-DMAC channel (channel 1) for ADC supports the reading of A/D conversion result from the ADC. This channel performs data transfer from the ADR0 register that stores the A/D conversion result of the ADC to the alias area.

  Alias areas for ATU-III (timer G) and ADC channels can be specified by ADMAAR and ADMATCR.

Figure 12.3 shows an overview of the DMA transfer.

- Transfer request
  - Channel 0: Timer G2 compare match
  - Channel 1: ADC_A and AN0 interrupt conversion end of ADC
- Addressing

  Addressing in registers is fixed (not incremented or decremented). Addressing in alias area is post-incremented (incremented by one in channel 0 and incremented by two in channel 1) each transfer.

  If the next transfer is performed while ADMATCR = 1 and ADMARTCR ≠ 0, ADMATCR and ADMAAR are updated by the corresponding reload register values. Accordingly, by setting ADMAAR = ADMARAR and ADMATCR = ADMARTCR, the data structure of the ring list (ring buffer) can be configured in the alias area.
- Transfer flow

  Data transfer is performed in the following sequence. Figure 12.4 shows the flowchart of data transfer.

  1. Sets reload registers for ADMATCR and ADMAAR.
  2. Specifies the start address of the alias area by ADMAAR and specifies the transfer count by ADMATCR.
  3. Performs one transfer if a transfer request occurs while transfer is enabled (DME = 1, NMIF = 0, ADMATCR ≠ 0), decrements ADMATCR, and updates ADMAAR.
  4. Transfers reload register values to ADMATCR and ADMAAR if data transfer is performed while ADMATCR = 1 and ADMARTCR ≠ 0.
  5. Suspends data transfer if an NMI interrupt occurs or if the DME bit is cleared to 0. Disables data transfer if 0 is written to ADMATCR while ADMATCR ≠ 0.
- Notes

  Reload registers ADMARAR and ADMARTCR for ADMAAR and ADMATCR must be set before ADMAAR and ADMATCR are set. Otherwise, reloading may not be performed correctly.

  In the alias area used as a transfer source of channel 0, the data for ADC input signal switching must be specified by the upper 4 bits in each byte.

  In the alias area used for channel 1, the contents in the alias area used as a transfer destination are updated with the address register whenever a transfer is requested. Accordingly, the contents of the alias area will be overwritten before they are read.

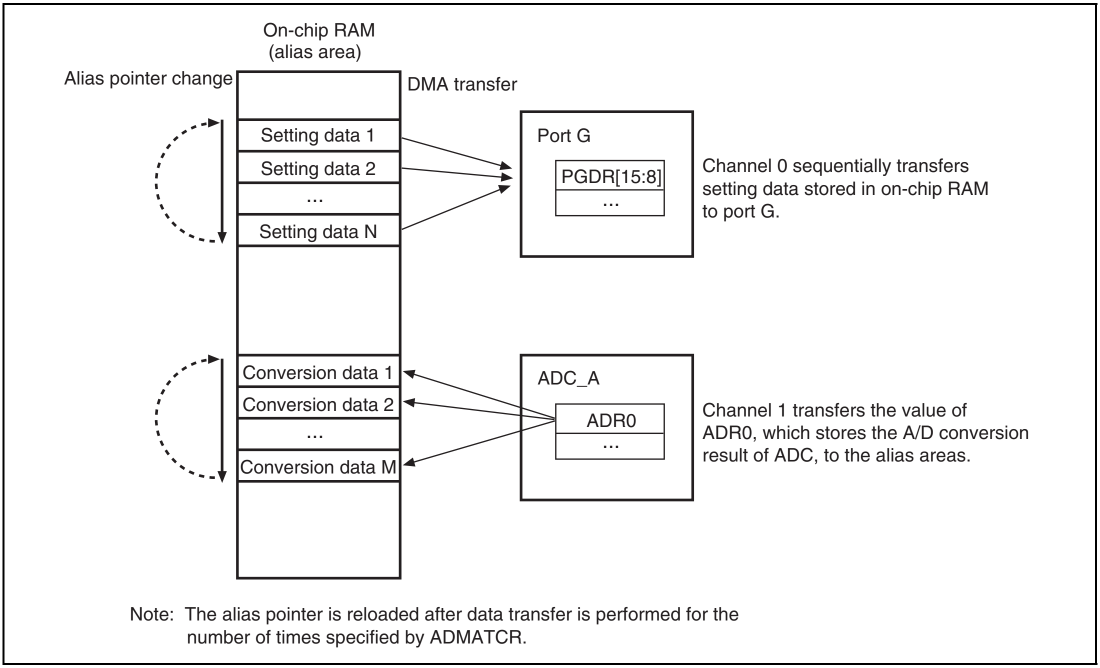
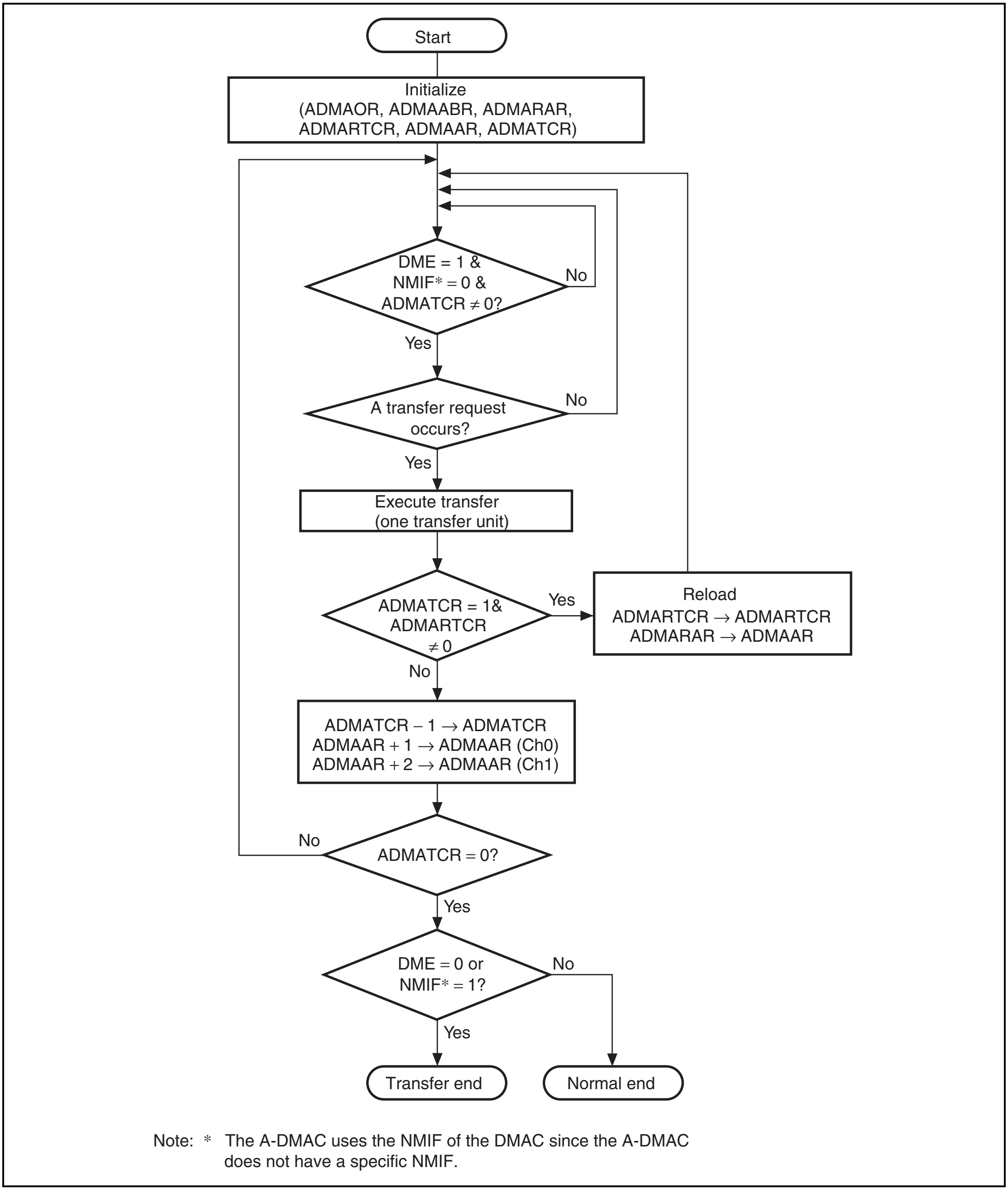

On-chip RAM
(alias area)
Alias pointer change
DMA transfer
Setting data 1
Setting data 2
...
Setting data N
Port G
PGDR[15:8]
...
Channel 0 sequentially transfers setting data stored in on-chip RAM to port G.
Conversion data 1
Conversion data 2
...
Conversion data M
ADC_A
ADR0
...
Channel 1 transfers the value of ADR0, which stores the A/D conversion result of ADC, to the alias areas.
Note: The alias pointer is reloaded after data transfer is performed for the number of times specified by ADMATCR.

Figure 12.3 Transfer Overview (Channels for ATU-III (timer G) and ADC)



Note: * The A-DMAC uses the NMIF of the DMAC since the A-DMAC does not have a specific NMIF.

Figure 12.4 Flowchart of DMA Transfer (Channels for ATU-III (timer G) and ADC)

(2) Operation for A-DMAC Channels Used for ATU-III (timers A, C, and F)

- Overview

  A-DMAC channels for the ATU-III (timers A, C, and F) perform data transfer from ATU-III registers to the alias areas. The alias area size for each channel is defined as follows.

  — A-DMAC channel for timer A: 8 bytes (area for 32-bit register × 2)

  — A-DMAC channel for timer C: 4 bytes

  — A-DMAC channel for timer F: 8 bytes (area for 32-bit register × 2)

  Refer to table 12.15, for details on the relationship between transfer source register name and transfer destination address. Figure 12.5 shows an overview of the DMA transfer.

- Transfer request

  An input capture interrupt of the corresponding ATU-III channel is used as a transfer request.

- Addressing

  Both transfer source and destination addresses are fixed (not incremented or decremented).

- Transfer flow

  Data transfer is performed in the following sequence. Figure 12.6 shows the flowchart of data transfer.

  1. Performs one transfer if a transfer request occurs while transfer is enabled (DE =1, DME = 1, NMIF = 0), and sets the DV bit to 1. In this case, if an interrupt is enabled (IE = 1), requests a transfer end interrupt to the CPU.
  2. Performs one transfer each time a transfer request occurs while transfer is enabled.
  3. Disables transfer if an NMI interrupt occurs, or if the DME or DE bit is cleared to 0.

- Notes

  In transfer enable state, the alias area specified as transfer destination is overwritten each time a transfer is requested and the alias area always indicates the latest register value.

  In A-DMAC channel for timer A, input capture value to be read for a current transfer request and ADMABUF register storage value (= input capture value read in the previous transfer request) are transferred to the alias area. (For details, refer to section 12.3.4 (1), Supplementary Description for A-DMAC Channels Used for ATU-III (timer A).)

  In A-DMAC channel for timer F, registers to be transferred differ depending on the corresponding ADMAMODE settings. If the ADMAMODE bit is cleared to 0 (default), the CDRF registers are transferred to the upper 4 bytes of alias area for each channel. While, if the ADMAMODE bit is set to 1, the CDRF and GRCF registers are transferred to the upper and lower 4 bytes of each alias area, respectively.

Table 12.15 A-DMAC Channels for ATU-III (Timers A, C, and F)

Channel No.	Transfer Request Source			Transfer Register	Relative Addresses in Alias Area
2	ATU-III (Timer A)	Channel 0		ICRA0	H'000 to H'007
3		Channel 1		ICRA1	H'008 to H'00F
4		Channel 2		ICRA2	H'010 to H'017
5		Channel 3		ICRA3	H'018 to H'01F
6		Channel 4		ICRA4	H'020 to H'027
7		Channel 5		ICRA5	H'028 to H'02F
8	ATU-III (Timer C)	Subblock C0	Channel 0	GRC00	H'030 to H'033
9			Channel 1	GRC01	H'034 to H'037
10			Channel 2	GRC02	H'038 to H'03B
11			Channel 3	GRC03	H'03C to H'03F
12		Subblock C1	Channel 0	GRC10	H'040 to H'043
13			Channel 1	GRC11	H'044 to H'047
14			Channel 2	GRC12	H'048 to H'04B
15			Channel 3	GRC13	H'04C to H'04F
16		Subblock C2	Channel 0	GRC20	H'050 to H'053
17			Channel 1	GRC21	H'054 to H'057
18			Channel 2	GRC22	H'058 to H'05B
19			Channel 3	GRC23	H'05C to H'05F
20		Subblock C3	Channel 0	GRC30	H'060 to H'063
21			Channel 1	GRC31	H'064 to H'067
22			Channel 2	GRC32	H'068 to H'06B
23			Channel 3	GRC33	H'06C to H'06F
24		Subblock C4	Channel 0	GRC40	H'070 to H'073
25			Channel 1	GRC41	H'074 to H'077
26			Channel 2	GRC42	H'078 to H'07B
27			Channel 3	GRC43	H'07C to H'07F

Channel No.	Transfer Request Source		Transfer Register	Relative Addresses in Alias Area
28	ATU-III (Timer F)	Channel 0	CDRF00, GRCF00	H'080 to H'087
29		Channel 1	CDRF01, GRCF01	H'088 to H'08F
30		Channel 2	CDRF02, GRCF02	H'090 to H'097
31		Channel 3	CDRF03, GRCF03	H'098 to H'09F
32		Channel 4	CDRF04, GRCF04	H'0A0 to H'0A7
33		Channel 5	CDRF05, GRCF05	H'0A8 to H'0AF
34		Channel 6	CDRF06, GRCF06	H'0B0 to H'0B7
35		Channel 7	CDRF07, GRCF07	H'0B8 to H'0BF
36		Channel 8	CDRF08, GRCF08	H'0C0 to H'0C7
37		Channel 9	CDRF09, GRCF09	H'0C8 to H'0CF
38		Channel 10	CDRF10, GRCF10	H'0D0 to H'0D7
39		Channel 11	CDRF11, GRCF11	H'0D8 to H'0DF
40		Channel 12	CDRF12, GRCF12	H'0E0 to H'0E7
41		Channel 13	CDRF13, GRCF13	H'0E8 to H'0EF
42		Channel 14	CDRF14, GRCF14	H'0F0 to H'0F7
43		Channel 15	CDRF15, GRCF15	H'0F8 to H'0FF
44		Channel 16	CDRF16, GRCF16	H'100 to H'107
45		Channel 17	CDRF17, GRCF17	H'108 to H'10F
46		Channel 18	CDRF18, GRCF18	H'110 to H'117
47		Channel 19	CDRF19, GRCF19	H'118 to H'11F
48		Channel 20	CDRF20, GRCF20	H'120 to H'127
49		Channel 21	CDRF21, GRCF21	H'128 to H'12F
50		Channel 22	CDRF22, GRCF22	H'130 to H'137
51		Channel 23	CDRF23, GRCF23	H'138 to H'13F
52		Channel 24	CDRF24, GRCF24	H'140 to H'147
53		Channel 25	CDRF25, GRCF25	H'148 to H'14F
54		Channel 26	CDRF26, GRCF26	H'150 to H'157
55		Channel 27	CDRF27, GRCF27	H'158 to H'15F

(3) Operation for A-DMAC Channels Used for RSPI and SCI

- Overview

  A-DMAC channels for the RSPI and SCI supports the following data transfer. In even channels, data transfer from a register to the alias area is performed; while in odd channels, data transfer from the alias area to a register is performed. Figure 12.5 shows an overview of the DMA transfer.

  The alias area size for each channel is obtained by (read (write) register size) × (transfer count specified by ADMATCR). Maximum alias sizes for RSPI channels and SCI channels are 2046 bytes and 1023 bytes, respectively. (For details, refer to section 12.2.7, A-DMAC Transfer Mode Registers (ADMAMODE), and section 12.2.9, A-DMAC Alias Pointer Registers (ADMAAR).)

  Table 12.16 summarizes each channel transfer request, transfer source register name, and transfer direction.

- Transfer request

  An RSPI or SCI reception or transmission interrupt for each channel is used as a transfer request. In transfer enable state, one data transfer is performed for a transfer request.

  In even channels for RSPI, a request generated at RSPI receive buffer full is used as a DMA transfer request. In odd channels for the RSPI, a request generated at RSPI transmit buffer empty is used as a DMA request.

  In even channels for the SCI, a request generated at SCI receive data full is used as a DMA transfer request. In odd channels for the SCI, a request generated at SCI transmit data empty is used as a DMA request.

  The transfer enable state is entered if the transfer count is specified in ADMATCR corresponding to each channel, if the TE bit is cleared to 0 (only when the TE bit is set to 1), and if the DME bit in ADMAOR is set to 1. (For details, refer to section 12.3.6, Transfer Enable/Disable Conditions and Interrupt Requests.)

- Addressing

  Addressing in registers is fixed (not incremented or decremented). Addressing in alias area is post-incremented (incremented by two in channels for RSPI and incremented by one in channels for the SCI) each transfer.

- Transfer flow

  Data transfer is performed in the following sequence. Figure 12.6 shows the flowchart of data transfer.

  1. Specifies the start address of the alias area by ADMAAR and specifies the transfer count by ADMATCR.
  2. Performs one transfer if a transfer request occurs while transfer is enabled (DME = 1, NMIF = 0, ADMATCR ≠ 0, TE = 0), decrements ADMATCR, and increments ADMAAR.

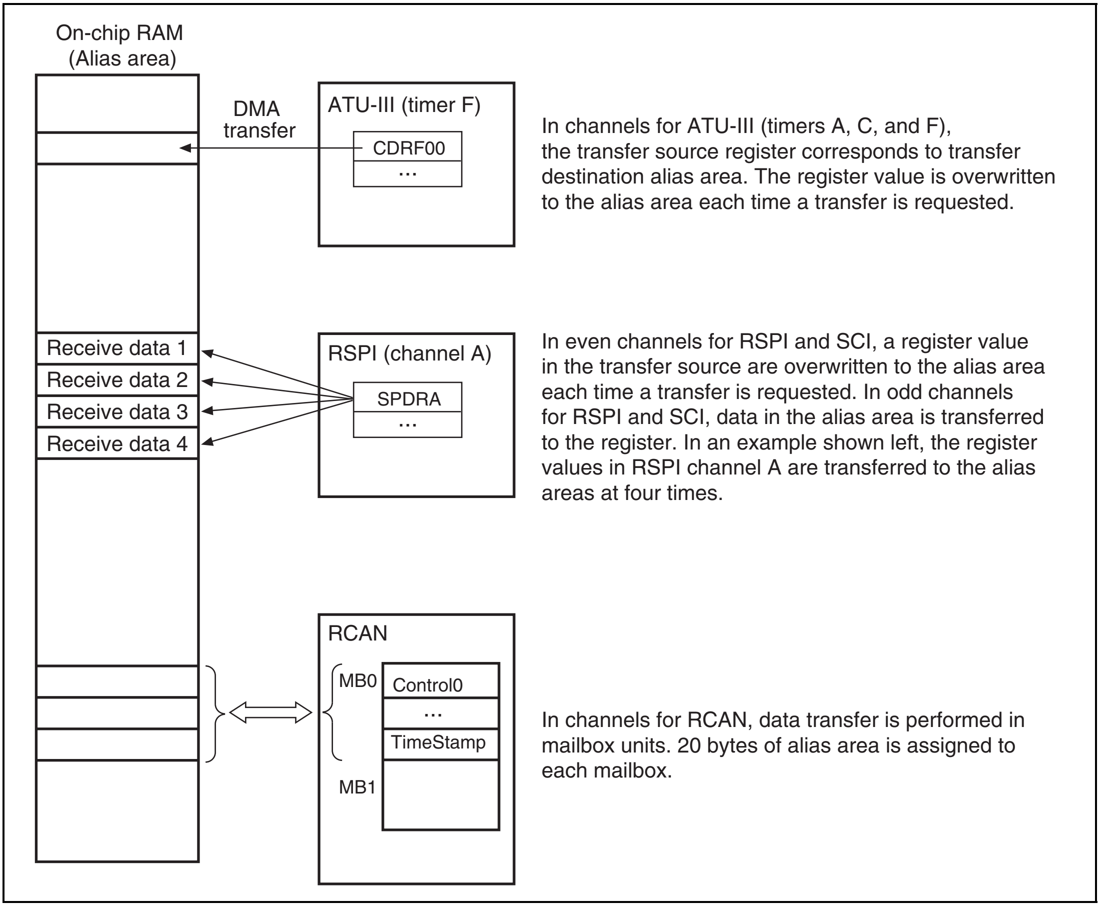
3. Sets the TE bit to 1 if ADMATCR is cleared 0 and if data transfer of specified count has been performed, and requests an interrupt to the CPU if an interrupt is enabled (IE = 1).
4. Suspends data transfer if an NMI interrupt occurs or if the DME bit is cleared to 0. Disables data transfer if 0 is written to ADMATCR while ADMATCR ≠ 0.

- Notes

  To re-enable data transfer while data transfer has been completed (TE = 1, ADMATCR = 0), ADMATCR must be set before the TE bit is cleared to 0. If the TE bit is cleared to 0 first, the transfer request may be directly informed to the interrupt controller. (For details, refer to section 12.3.6, Transfer Enable/Disable Conditions and Interrupt Requests.)

Table 12.16 Channels for RSPI and SCI

Channel No.	Transfer Request Source		Transfer Request Signal	Transfer Register	Transfer Direction	Transfer Bytes
56	RSPI	Channel A	Receive buffer full (SPRI)	SPDRA	IO → RAM	2
57			Transmit buffer empty (SPTI)	SPDRA	IO ← RAM	2
58		Channel B	Receive buffer full (SPRI)	SPDRB	IO → RAM	2
59			Transmit buffer empty (SPTI)	SPDRB	IO ← RAM	2
60		Channel C	Receive buffer full (SPRI)	SPDRC	IO → RAM	2
61			Transmit buffer empty (SPTI)	SPDRC	IO ← RAM	2
62	SCI	Channel A	Receive data full	SCRDR1A	IO → RAM	1
63			Transmit data empty	SCTDR1A	IO ← RAM	1
64		Channel B	Receive data full	SCRDR1B	IO → RAM	1
65			Transmit data empty	SCTDR1B	IO ← RAM	1
66		Channel C	Receive data full	SCRDR1C	IO → RAM	1
67			Transmit data empty	SCTDR1C	IO ← RAM	1
68		Channel D	Receive data full	SCRDR1D	IO → RAM	1
69			Transmit data empty	SCTDR1D	IO ← RAM	1
70		Channel E	Receive data full	SCRDR1E	IO → RAM	1
71			Transmit data empty	SCTDR1E	IO ← RAM	1

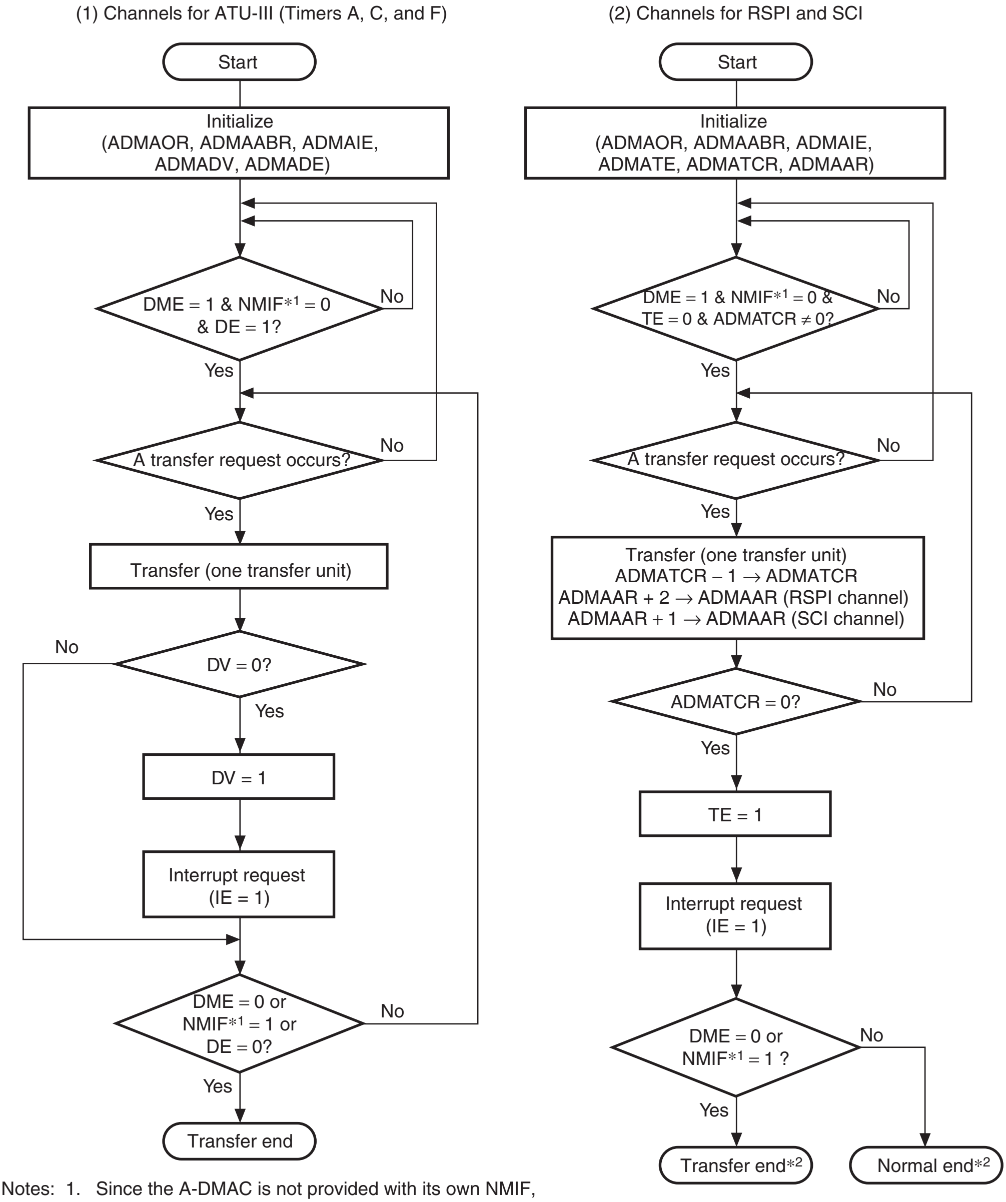


In channels for ATU-III (timers A, C, and F), the transfer source register corresponds to transfer destination alias area. The register value is overwritten to the alias area each time a transfer is requested.

In even channels for RSPI and SCI, a register value in the transfer source are overwritten to the alias area each time a transfer is requested. In odd channels for RSPI and SCI, data in the alias area is transferred to the register. In an example shown left, the register values in RSPI channel A are transferred to the alias areas at four times.

In channels for RCAN, data transfer is performed in mailbox units. 20 bytes of alias area is assigned to each mailbox.

Figure 12.5 Transfer Overview (Channels for ATU-III (Timers A, C, and F), RSPI, SCI, and RCAN)


(1) Channels for ATU-III (Timers A, C, and F)
Start
Initialize (ADMAOR, ADMAABR, ADMAIE, ADMADV, ADMADE)
DME = 1 & NMIF*1 = 0 & DE = 1?
No
Yes
A transfer request occurs?
No
Yes
Transfer (one transfer unit)
DV = 0?
No
Yes
DV = 1
Interrupt request (IE = 1)
DME = 0 or NMIF*1 = 1 or DE = 0?
No
Yes
Transfer end
(2) Channels for RSPI and SCI
Start
Initialize (ADMAOR, ADMAABR, ADMAIE, ADMATE, ADMATCR, ADMAAR)
DME = 1 & NMIF*1 = 0 & TE = 0 & ADMATCR ≠ 0?
No
Yes
A transfer request occurs?
No
Yes
Transfer (one transfer unit)
ADMATCR – 1 → ADMATCR
ADMAAR + 2 → ADMAAR (RSPI channel)
ADMAAR + 1 → ADMAAR (SCI channel)
ADMATCR = 0?
No
Yes
TE = 1
Interrupt request (IE = 1)
DME = 0 or NMIF*1 = 1 ?
No
Yes
Transfer end*2
Normal end*2

Notes: 1. Since the A-DMAC is not provided with its own NMIF, it uses the NMIF of the DMAC instead.
2. To re-enable data transfer, be sure to set ADMATCR before clearing the TE bit to 0.

Figure 12.6 Flowchart of DMA Transfer (Channels for ATU-III (Timers A, C, and F), RSPI, and SPI)

(4) Operation for A-DMAC Channels Used for RCAN

- Overview

  A-DMAC channels for the RCAN performs data transfer between mailboxes in the RCAN module and alias areas. In channel 72 and channel 73, a receive operation is performed (data is transferred from a mailbox (MBx) to the alias area). In channel 74 and channel 75, a transmit operation is performed (data is transferred from alias area to the mailbox (MBx)). Figure 12.5 shows an overview of the DMA transfer.

  Table 12.17 shows the correspondence between mailboxes (MBx) and alias areas. The alias area for each mailbox (MBx) is assigned in 32-byte units. Receive alias areas are common to transmit alias areas.

  For one mailbox (MBx) transfer, data is transferred in long-word units at five times and a total of 20-byte data is transferred. Fields to be transferred are ID (4 bytes), LAFM (4 bytes), Data (8 bytes), Control 1 (2 bytes), and TimeStamp (2 bytes).

Table 12.17 Correspondence between Mailbox (MBx) and Alias Area Addresses

MBx	RCAN_A	RCAN_B	RCAN_C	RCAN_D
0	H'0200 to H'0213	H'0600 to H'0613	H'0A00 to H'0A13	H'0E00 to H'0E13
1	H'0220 to H'0233	H'0620 to H'0633	H'0A20 to H'0A33	H'0E20 to H'0E33
2	H'0240 to H'0253	H'0640 to H'0653	H'0A40 to H'0A53	H'0E40 to H'0E53
3	H'0260 to H'0273	H'0660 to H'0673	H'0A60 to H'0A73	H'0E60 to H'0E73
4	H'0280 to H'0293	H'0680 to H'0693	H'0A80 to H'0A93	H'0E80 to H'0E93
5	H'02A0 to H'02B3	H'06A0 to H'06B3	H'0AA0 to H'0AB3	H'0EA0 to H'0EB3
6	H'02C0 to H'02D3	H'06C0 to H'06D3	H'0AC0 to H'0AD3	H'0EC0 to H'0ED3
7	H'02E0 to H'02F3	H'06E0 to H'06F3	H'0AE0 to H'0AF3	H'0EE0 to H'0EF3
8	H'0300 to H'0313	H'0700 to H'0713	H'0B00 to H'0B13	H'0F00 to H'0F13
9	H'0320 to H'0333	H'0720 to H'0733	H'0B20 to H'0B33	H'0F20 to H'0F33
10	H'0340 to H'0353	H'0740 to H'0753	H'0B40 to H'0B53	H'0F40 to H'0F53
11	H'0360 to H'0373	H'0760 to H'0773	H'0B60 to H'0B73	H'0F60 to H'0F73
12	H'0380 to H'0393	H'0780 to H'0793	H'0B80 to H'0B93	H'0F80 to H'0F93
13	H'03A0 to H'03B3	H'07A0 to H'07B3	H'0BA0 to H'0BB3	H'0FA0 to H'0FB3
14	H'03C0 to H'03D3	H'07C0 to H'07D3	H'0BC0 to H'0BD3	H'0FC0 to H'0FD3
15	H'03E0 to H'03F3	H'07E0 to H'07F3	H'0BE0 to H'0BF3	H'0FE0 to H'0FF3
16	H'0400 to H'0413	H'0800 to H'0813	H'0C00 to H'0C13	H'1000 to H'1013
17	H'0420 to H'0433	H'0820 to H'0833	H'0C20 to H'0C33	H'1020 to H'1033
18	H'0440 to H'0453	H'0840 to H'0853	H'0C40 to H'0C53	H'1040 to H'1053
19	H'0460 to H'0473	H'0860 to H'0873	H'0C60 to H'0C73	H'1060 to H'1073
20	H'0480 to H'0493	H'0880 to H'0893	H'0C80 to H'0C93	H'1080 to H'1093
21	H'04A0 to H'04B3	H'08A0 to H'08B3	H'0CA0 to H'0CB3	H'10A0 to H'10B3
22	H'04C0 to H'04D3	H'08C0 to H'08D3	H'0CC0 to H'0CD3	H'10C0 to H'10D3
23	H'04E0 to H'04F3	H'08E0 to H'08F3	H'0CE0 to H'0CF3	H'10E0 to H'10F3
24	H'0500 to H'0513	H'0900 to H'0913	H'0D00 to H'0D13	H'1100 to H'1113
25	H'0520 to H'0533	H'0920 to H'0933	H'0D20 to H'0D33	H'1120 to H'1133
26	H'0540 to H'0553	H'0940 to H'0953	H'0D40 to H'0D53	H'1140 to H'1153
27	H'0560 to H'0573	H'0960 to H'0973	H'0D60 to H'0D73	H'1160 to H'1173
28	H'0580 to H'0593	H'0980 to H'0993	H'0D80 to H'0D93	H'1180 to H'1193
29	H'05A0 to H'05B3	H'09A0 to H'09B3	H'0DA0 to H'0DB3	H'11A0 to H'11B3
30	H'05C0 to H'05D3	H'09C0 to H'09D3	H'0DC0 to H'0DD3	H'11C0 to H'11D3
31	H'05E0 to H'05F3	H'09E0 to H'09F3	H'0DE0 to H'0DF3	H'11E0 to H'11F3

Note: Only the lower addresses are indicated in hexadecimal.

- Transfer request

  A channel for reception uses the MBx full information from the RCAN uses a transfer request. The mailbox (MBx) to be transferred is detected by scanning the mailbox (MBx) status.

  A channel for transmission is starts transfer by software.

- Addressing

  Addressing in both transfer source and destination are incremented.

- Operation Details and Transfer flow

  Channel 72 and channel 73: One mailbox (MBx) transfer is performed by a transfer request. The mailbox (MBx) number to be transferred is informed by the RCAN synchronously with the transfer request.

  When the mailbox (MBx) transfer is completed, the RV bit corresponding to the mailbox (MBx) is set to 1. Even if the mailbox (MBx) transfer whose RV bit is set to 1 is requested from the RCAN, the A-DMAC does not accept the transfer request. In this case, the A-DMAC informs the RCAN of the MBx transfer end and the RCAN clears the corresponding bit in the receive wait register (or frame wait register). Although an MBX transfer from RCAN_A to RCAN_C is performed on channel 72 when transfer on channel 73 is not enabled, MBx transfer from RCAN_A to RCAN_B proceeds when transfer from RCAN_C to RCAN_D on channel 73 has been enabled.

  Data transfer is performed in the following sequence. Figure 12.7 shows the flowchart of data transfer.

  A. Performs one mailbox (MBx) transfer if a transfer request occurs while transfer is enabled (DME = 1, NMIF = 0, DE = 1).

  B. Sets the RV bit corresponding to the mailbox (MBx) to 1.

  C. Suspends data transfer if an NMI interrupt occurs or if the DME bit or DE bit is cleared to 0.

     (For details, refer to section 12.3.3, Transfer Suspension and Resumption.) If a transfer request to the high priority DMA channel occurs during a mailbox (MBx) transfer, channel 72 enters transfer wait state.

  Channel 74 and channel 75: The number of mailbox (MBx) to be transferred to the transmit wait register is set by software. If the transfer enable state is then entered, the specified mailboxes are transferred according to the mailbox priority (Priority: RCAN_A-MBx1 → RCAN_A-MBx2 →…).

  Data transfer is performed in the following sequence. Figure 12.8 shows the flowchart of data transfer.

  1. Sets a bit in ADMATVPR corresponding to the number of mailbox and sets the DE bit to 1. If the DE bit is set to 1 without specifying mailbox (MBx), performs the operation in step 5 below.

2. Performs one mailbox (MBx) transfer corresponding to the TV bit while transfer is enabled (DME = 1, NMIF = 0, DE = 1, TE = 0).
3. Clears the TV bit corresponding to the mailbox (MBx) to be transferred to 0.
4. Repeats operations in steps 2 and 3 until all the TV bits are cleared to 0.
5. Sets the TE bit to 1 if all the TV bits are cleared to 0 and if the specified transfer has been completed. In this case, if an interrupt is enabled (IE = 1), an interrupt is requested to the CPU.
6. While the TE bit is set to 1, transfer is not performed even if the TV bit is set.
7. Suspends data transfer if an NMI interrupt occurs or if the DME bit or DE bit is cleared to 0.

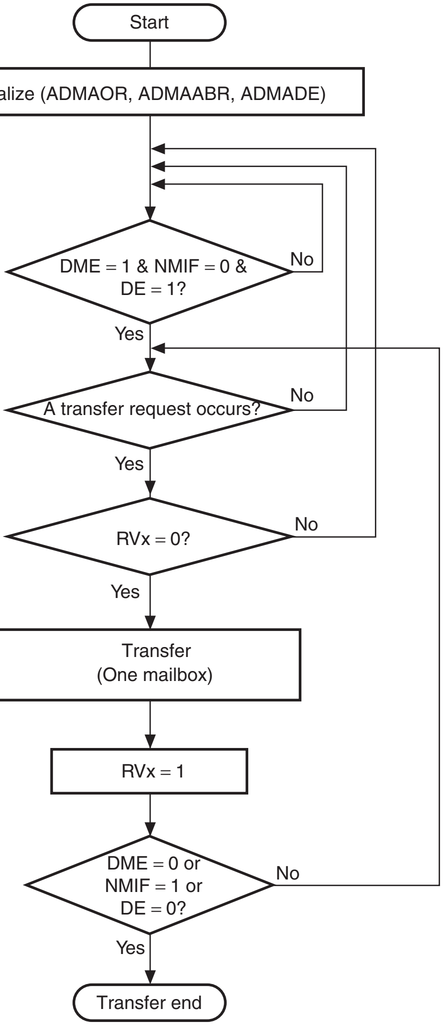
   (For details, refer to section 12.3.3, Transfer Suspension and Resumption.) If a transfer request to the high DMA channel occurs during a mailbox (MBx) transfer, channel 74 enters transfer wait state.

- Notes

  In a receive transfer, data transfer is enabled when the RV bit is cleared. Accordingly, the alias area must be referenced before the RV bit is cleared to avoid conflict.

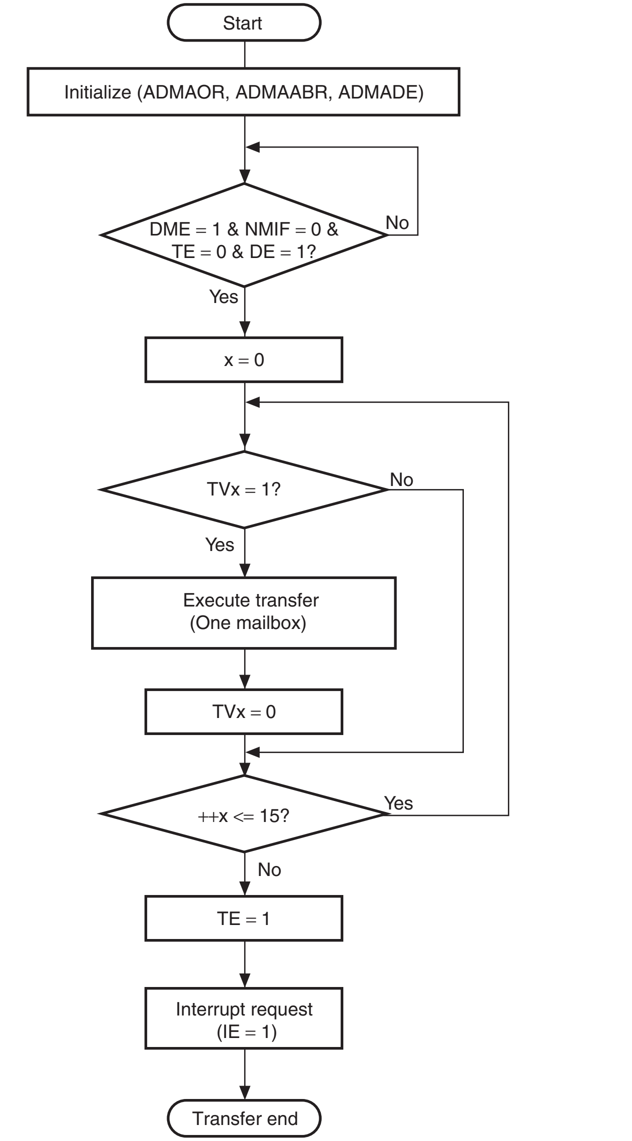
  Note that the order that MBx in which RCAN becomes full does not match the order in which MBx in RCAN is transferred to the alias area. This is because the A-DMAC does not accept a transfer in MBx units if the RV bit is set to 1 and because the algorithm that selects MBx is not a FIFO.

  In a transmit transfer, the TE bit is set to 1 without transferring data if transfer is enabled while all the TV bits are cleared to 0. A write to a TV bit is prohibited in the transfer enable state.


Start
Initialize (ADMAOR, ADMAABR, ADMADE)
DME = 1 & NMIF = 0 & DE = 1?
No
Yes
A transfer request occurs?
No
Yes
RVx = 0?
No
Yes
Transfer (One mailbox)
RVx = 1
DME = 0 or NMIF = 1 or DE = 0?
No
Yes
Transfer end

Note: This flowchart is simplified assuming that no suspension occurs during one mailbox (MBx) transfer.

Figure 12.7 DMA Transfer Flowchart (Channel for RCAN Reception)



Note: This flowchart is simplified by indicating the processing relating to TV0 to TV15 of the specific ADMATVPR register. In actual operation, TV0 to TV15 in all ADMATVPR registers are scanned.

Figure 12.8 DMA Transfer Flowchart (Channel for RCAN Transmission)

(5) Operation for FlexRay Channels

- Overview

  A channel for FlexRay performs the transfer of received data and data received with the FIFO from message RAM within the FlexRay module to the alias areas. Transfer on channel 76 is for a message buffer with reception settings and transfer on channel 77 is for a message buffer with reception FIFO settings. An overview of transfer is given in figure 12.9.

  On channels 76 and 77, both the header and data sections for each of the message buffers is transferred to the alias areas.

  The channel of the message buffer for transfer on channel 76 will vary according to the setting of the A-DMAC FlexRay control register (ADMAFRCTR). Tables 12.18, 12.20, 12.22, and 12.24 show the message buffer numbers for different payload length settings and the corresponding addresses of the alias areas that are the targets for transfer.

  The alias area corresponding to the message buffer set up for receive FIFO operation to be transferred on channel 77 become the same as for one channel of message buffer.

  The oldest data in the reception FIFO within message RAM are transferred to the alias area. Table 12.26 shows the address for corresponding alias areas to a transfer of receive FIFO.

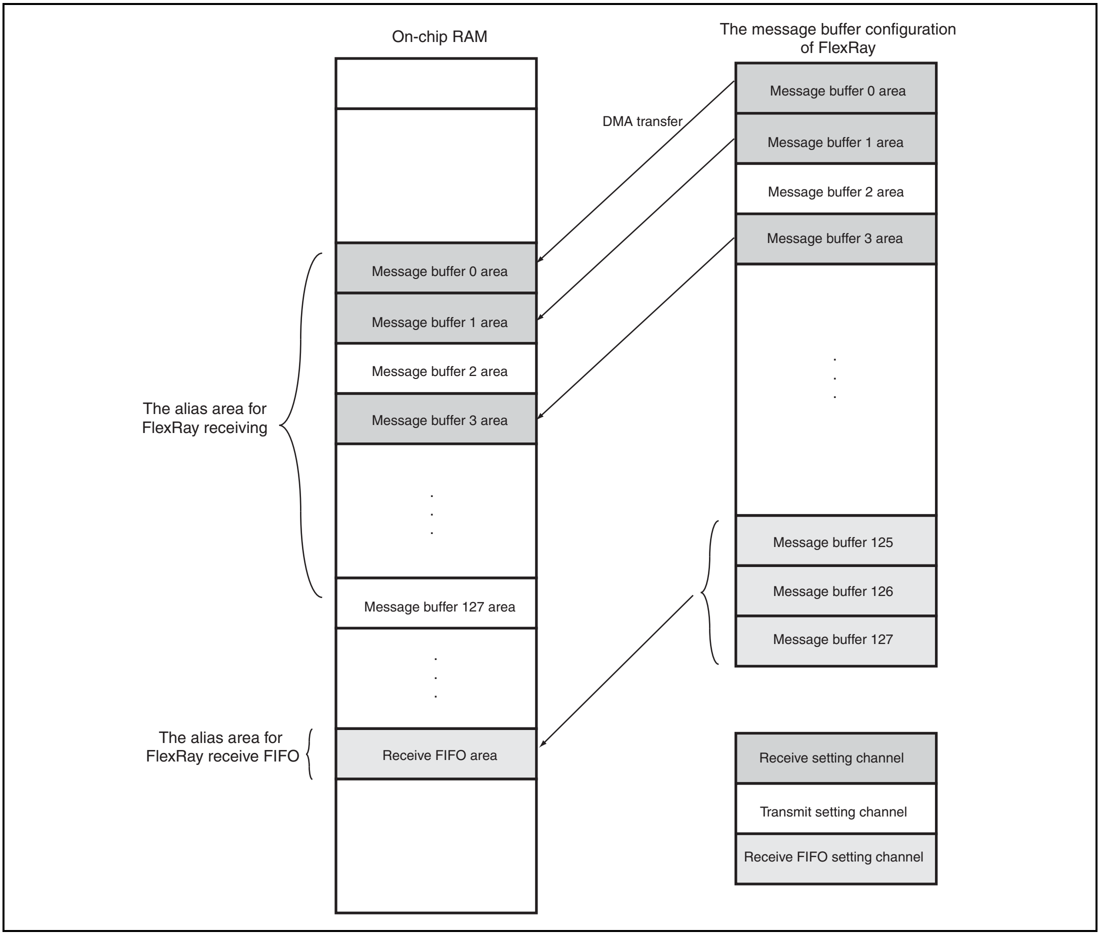


Figure 12.9 Overview of Transfer (Channels for FlexRay)

- Transfer on the channels for FlexRay proceeds in message-buffer units.
- The size of one message buffer for reception or reception-with-FIFO differs according to the reception-payload length setting.
- The number of message buffer areas for reception differs according to the reception-payload length setting.
- For channels with transmission settings, there is no transfer from message buffers of the FlexRay module to alias areas.
- The area of a message buffer for reception-with-FIFO is same as the portion for one message.

Table 12.18 Correspondence between message buffer numbers and the addresses of alias areas (when the payload length is set to 32 bytes)

Message buffer	Address	Message buffer	Address	Message buffer	Address	Message buffer	Address
0	H'1300 - H'132F	32	H'1900 - H'192F	64	H'1F00 - H'1F2F	96	H'2500 - H'252F
1	H'1330 - H'135F	33	H'1930 - H'195F	65	H'1F30 - H'1F5F	97	H'2530 - H'255F
2	H'1360 - H'138F	34	H'1960 - H'198F	66	H'1F60 - H'1F8F	98	H'2560 - H'258F
3	H'1390 - H'13BF	35	H'1990 - H'19BF	67	H'1F90 - H'1FBF	99	H'2590 - H'25BF
4	H'13C0 - H'13EF	36	H'19C0 - H'19EF	68	H'1FC0 - H'1FEF	100	H'25C0 - H'25EF
5	H'13F0 - H'141F	37	H'19F0 - H'1A1F	69	H'1FF0 - H'201F	101	H'25F0 - H'261F
6	H'1420 - H'144F	38	H'1A20 - H'1A4F	70	H'2020 - H'204F	102	H'2620 - H'264F
7	H'1450 - H'147F	39	H'1A50 - H'1A7F	71	H'2050 - H'207F	103	H'2650 - H'267F
8	H'1480 - H'14AF	40	H'1A80 - H'1AAF	72	H'2080 - H'20AF	104	H'2680 - H'26AF
9	H'14B0 - H'14DF	41	H'1AB0 - H'1ADF	73	H'20B0 - H'20DF	105	H'26B0 - H'26DF
10	H'14E0 - H'150F	42	H'1AE0 - H'1B0F	74	H'20E0 - H'210F	106	H'26E0 - H'270F
11	H'1510 - H'153F	43	H'1B10 - H'1B3F	75	H'2110 - H'213F	107	H'2710 - H'273F
12	H'1540 - H'156F	44	H'1B40 - H'1B6F	76	H'2140 - H'216F	108	H'2740 - H'276F
13	H'1570 - H'159F	45	H'1B70 - H'1B9F	77	H'2170 - H'219F	109	H'2770 - H'279F
14	H'15A0 - H'15CF	46	H'1BA0 - H'1BCF	78	H'21A0 - H'21CF	110	H'27A0 - H'27CF
15	H'15D0 - H'15FF	47	H'1BD0 - H'1BFF	79	H'21D0 - H'21FF	111	H'27D0 - H'27FF
16	H'1600 - H'162F	48	H'1C00 - H'1C2F	80	H'2200 - H'222F	112	H'2800 - H'282F
17	H'1630 - H'165F	49	H'1C30 - H'1C5F	81	H'2230 - H'225F	113	H'2830 - H'285F
18	H'1660 - H'168F	50	H'1C60 - H'1C8F	82	H'2260 - H'228F	114	H'2860 - H'288F
19	H'1690 - H'16BF	51	H'1C90 - H'1CBF	83	H'2290 - H'22BF	115	H'2890 - H'28BF
20	H'16C0 - H'16EF	52	H'1CC0 - H'1CEF	84	H'22C0 - H'22EF	116	H'28C0 - H'28EF
21	H'16F0 - H'171F	53	H'1CF0 - H'1D1F	85	H'22F0 - H'231F	117	H'28F0 - H'291F
22	H'1720 - H'174F	54	H'1D20 - H'1D4F	86	H'2320 - H'234F	118	H'2920 - H'294F
23	H'1750 - H'177F	55	H'1D50 - H'1D7F	87	H'2350 - H'237F	119	H'2950 - H'297F
24	H'1780 - H'17AF	56	H'1D80 - H'1DAF	88	H'2380 - H'23AF	120	H'2980 - H'29AF
25	H'17B0 - H'17DF	57	H'1DB0 - H'1DDF	89	H'23B0 - H'23DF	121	H'29B0 - H'29DF
26	H'17E0 - H'180F	58	H'1DE0 - H'1E0F	90	H'23E0 - H'240F	122	H'29E0 - H'2A0F
27	H'1810 - H'183F	59	H'1E10 - H'1E3F	91	H'2410 - H'243F	123	H'2A10 - H'2A3F
28	H'1840 - H'186F	60	H'1E40 - H'1E6F	92	H'2440 - H'246F	124	H'2A40 - H'2A6F
29	H'1870 - H'189F	61	H'1E70 - H'1E9F	93	H'2470 - H'249F	125	H'2A70 - H'2A9F
30	H'18A0 - H'18CF	62	H'1EA0 - H'1ECF	94	H'24A0 - H'24CF	126	H'2AA0 - H'2ACF
31	H'18D0 - H'18FF	63	H'1ED0 - H'1EFF	95	H'24D0 - H'24FF	127	H'2AD0 - H'2AFF

Note: Address are given in hexadecimal and shown only lower address.

Per message buffer transfers 12 times in longword units, totally 48 bytes are transferred. The data transferred are followings: FlexRay header section read register1 (FRRDHS1): 4bytes, FlexRay header section read register2 (FRRDHS2): 4bytes, FlexRay header section read register3 (FRRDHS3): 4bytes, FlexRay message buffer status register (FRMBS): 4bytes, FlexRay data section read register1 to 8 (FRRDS1 to 8): 32bytes.

Table 12.19 shows the data alignment in 1 message buffer of the alias areas.

Table 12.19 Data alignment (when payload length are set to 32 bytes)

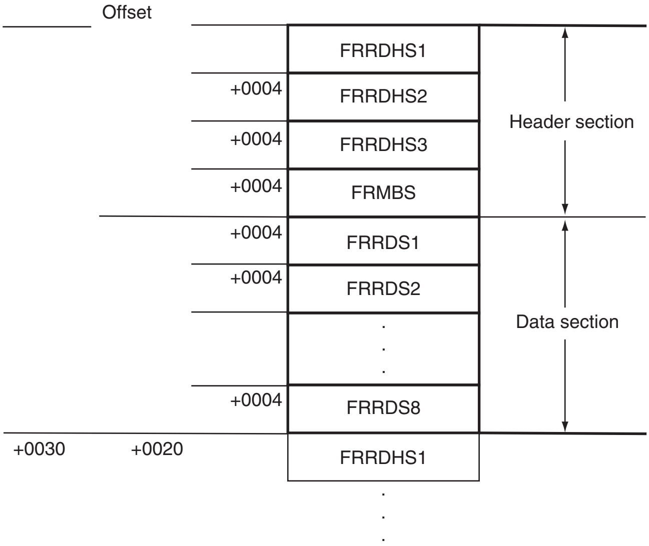


Table 12.20 The corresponding address of the message buffer and the alias area (when payload length are set to 32 bytes)

Message buffer	Address	Message buffer	Address
0	H'1300 - H'134F	38	H'1EE0 - H'1F2F
1	H'1350 - H'139F	39	H'1F30 - H'1F7F
2	H'13A0 - H'13EF	40	H'1F80 - H'1FCF
3	H'13F0 - H'143F	41	H'1FD0 - H'201F
4	H'1440 - H'148F	42	H'2020 - H'206F
5	H'1490 - H'14DF	43	H'2070 - H'20BF
6	H'14E0 - H'152F	44	H'20C0 - H'210F
7	H'1530 - H'157F	45	H'2110 - H'215F
8	H'1580 - H'15CF	46	H'2160 - H'21AF
9	H'15D0 - H'161F	47	H'21B0 - H'21FF
10	H'1620 - H'166F	48	H'2200 - H'224F
11	H'1670 - H'16BF	49	H'2250 - H'229F
12	H'16C0 - H'170F	50	H'22A0 - H'22EF
13	H'1710 - H'175F	51	H'22F0 - H'233F
14	H'1760 - H'17AF	52	H'2340 - H'238F
15	H'17B0 - H'17FF	53	H'2390 - H'23DF
16	H'1800 - H'184F	54	H'23E0 - H'242F
17	H'1850 - H'189F	55	H'2430 - H'247F
18	H'18A0 - H'18EF	56	H'2480 - H'24CF
19	H'18F0 - H'193F	57	H'24D0 - H'251F
20	H'1940 - H'198F	58	H'2520 - H'256F
21	H'1990 - H'19DF	59	H'2570 - H'25BF
22	H'19E0 - H'1A2F	60	H'25C0 - H'260F
23	H'1A30 - H'1A7F	61	H'2610 - H'265F
24	H'1A80 - H'1ACF	62	H'2660 - H'26AF
25	H'1AD0 - H'1B1F	63	H'26B0 - H'26FF
26	H'1B20 - H'1B6F	64	H'2700 - H'274F
27	H'1B70 - H'1BBF	65	H'2750 - H'279F
28	H'1BC0 - H'1C0F	66	H'27A0 - H'27EF
29	H'1C10 - H'1C5F	67	H'27F0 - H'283F
30	H'1C60 - H'1CAF	68	H'2840 - H'288F
31	H'1CB0 - H'1CFF	69	H'2890 - H'28DF
32	H'1D00 - H'1D4F	70	H'28E0 - H'292F
33	H'1D50 - H'1D9F	71	H'2930 - H'297F
34	H'1DA0 - H'1DEF	72	H'2980 - H'29CF
35	H'1DF0 - H'1E3F	73	H'29D0 - H'2A1F
36	H'1E40 - H'1E8F	74	H'2A20 - H'2A6F
37	H'1E90 - H'1EDF	75	H'2A70 - H'2ABF

Note: Address are given in hexadecimal and shown only lower address.

Per message buffer transfers 20 times in longword units, totally 80 bytes are transferred. The data transferred are followings: FlexRay header section read register1 (FRRDHS1): 4bytes, FlexRay header section read register2 (FRRDHS2): 4bytes, FlexRay header section read register3 (FRRDHS3): 4bytes, FlexRay message buffer status register (FRMBS): 4bytes, FlexRay data section read register1 to 16 (FRRDS1 to 16): 64bytes.

Table 12.21 shows the data alignment in 1 message buffer of the alias areas.

Table 12.21 Data alignment (when payload length are set to 64 bytes)

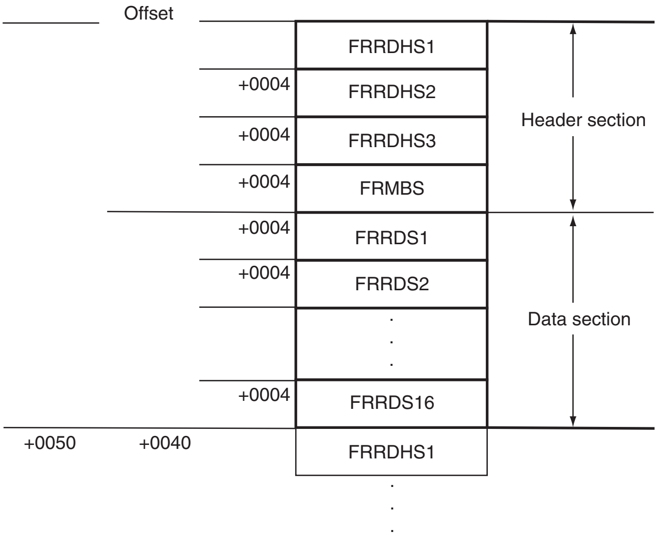


Table 12.22 The corresponding address of the message buffer and the alias area (when payload length are set to 128 bytes)

Message buffer	Address	Message buffer	Address
0	H'1300 - H'138F	21	H'1ED0 - H'1F5F
1	H'1390 - H'141F	22	H'1F60 - H'1FEF
2	H'1420 - H'14AF	23	H'1FF0 - H'207F
3	H'14B0 - H'153F	24	H'2080 - H'210F
4	H'1540 - H'15CF	25	H'2110 - H'219F
5	H'15D0 - H'165F	26	H'21A0 - H'222F
6	H'1660 - H'16EF	27	H'2230 - H'22BF
7	H'16F0 - H'177F	28	H'22C0 - H'234F
8	H'1780 - H'180F	29	H'2350 - H'23DF
9	H'1810 - H'189F	30	H'23E0 - H'246F
10	H'18A0 - H'192F	31	H'2470 - H'24FF
11	H'1930 - H'19BF	32	H'2500 - H'258F
12	H'19C0 - H'1A4F	33	H'2590 - H'261F
13	H'1A50 - H'1ADF	34	H'2620 - H'26AF
14	H'1AE0 - H'1B6F	35	H'26B0 - H'273F
15	H'1B70 - H'1BFF	36	H'2740 - H'27CF
16	H'1C00 - H'1C8F	37	H'27D0 - H'285F
17	H'1C90 - H'1D1F	38	H'2860 - H'28EF
18	H'1D20 - H'1DAF	39	H'28F0 - H'297F
19	H'1DB0 - H'1E3F	40	H'2980 - H'2A0F
20	H'1E40 - H'1ECF	41	H'2A10 - H'2A9F

Note: Addresses are given in hexadecimal and shown only lower address.

Per message buffer transfers 36 times in longword units, totally 144 bytes are transferred. The data transferred are followings: FlexRay header section read register1 (FRRDHS1): 4bytes, FlexRay header section read register2 (FRRDHS2): 4bytes, FlexRay header section read register3 (FRRDHS3): 4bytes, FlexRay message buffer status register (FRMBS): 4bytes, FlexRay data section read register1 to 32 (FRRDS1 to 32): 128bytes.

Table 12.23 shows the data alignment in 1 message buffer of the alias areas.

Table 12.23 Data alignment (when payload length are set to 128 bytes)

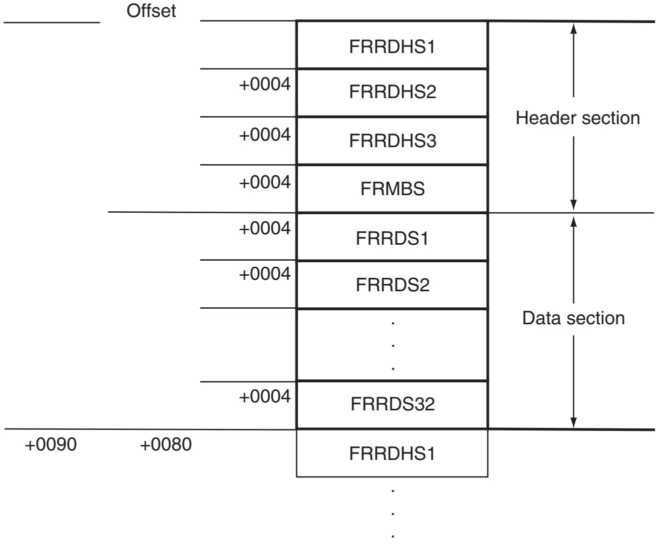


Table 12.24 The corresponding address of the message buffer and the alias area (when payload length are set to 254 bytes)

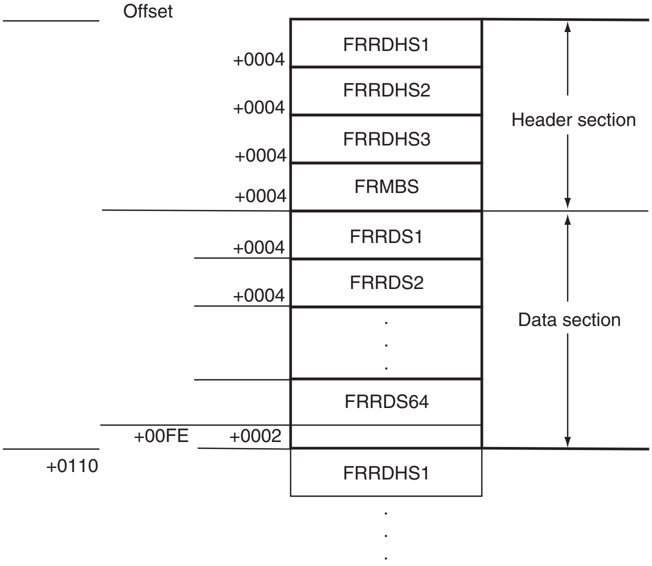
Message buffer	Address
0	H'1300 - H'140F
1	H'1410 - H'151F
2	H'1520 - H'162F
3	H'1630 - H'173F
4	H'1740 - H'184F
5	H'1850 - H'195F
6	H'1960 - H'1A6F
7	H'1A70 - H'1B7F
8	H'1B80 - H'1C8F
9	H'1C90 - H'1D9F
10	H'1DA0 - H'1EAF
11	H'1EB0 - H'1FBF
12	H'1FC0 - H'20CF
13	H'20D0 - H'21DF
14	H'21E0 - H'22EF
15	H'22F0 - H'23FF
16	H'2400 - H'250F
17	H'2510 - H'261F
18	H'2620 - H'272F
19	H'2730 - H'283F
20	H'2840 - H'294F
21	H'2950 - H'2A5F

Note: Addresses are given in hexadecimal and shown only lower address.

Per message buffer transfers 68 times in longword units, totally 272 bytes are transferred. The data transferred are followings: FlexRay header section read register1 (FRRDHS1): 4bytes, FlexRay header section read register2 (FRRDHS2): 4bytes, FlexRay header section read register3 (FRRDHS3): 4bytes, FlexRay message buffer status register (FRMBS): 4bytes, FlexRay data section read register1 to 64 (FRRDS1 to 64): 254bytes.

Table 12.25 shows the data alignment in 1 message buffer of the alias areas.

Table 12.25 Data alignment (when payload length are set to 254 bytes)



Note: The alias areas for 1 message buffer is assigned 272 bytes, however the total size of the header section and data section are 270 bytes. H'0000 are transferred to the last half of 2 bytes.

Table 12.26 The corresponding address of the message buffer and the alias areas

When payload length is 32 bytes

Message buffer	Address
The oldest FIFO data	H'2B10 to H'2B3F

When payload length is 64 bytes

Message buffer	Address
The oldest FIFO data	H'2B10 to H'2B5F

When payload length is 128 bytes

Message buffer	Address
The oldest FIFO data	H'2B10 to H'2B9F

When payload length is 254 bytes

Message buffer	Address
The oldest FIFO data	H'2B10 to H'2C1F

Transferring to the alias areas for message buffer of receive FIFO varies depending on the settings of A-DMAC FlexRay FIFO Control Register (ADMAFRFCTR). When payload length is 32bytes, per message buffer transfers 12 times in longword units, totally 48 bytes are transferred. The data transferred are followings: FlexRay header section read register1 (FRRDHS1): 4bytes, FlexRay header section read register2 (FRRDHS2): 4bytes, FlexRay header section read register3 (FRRDHS3): 4bytes, FlexRay message buffer status register (FRMBS): 4bytes, FlexRay data section read register1 to 8 (FRRDS1 to 8): 32bytes.

The message buffer data alignment on the alias areas are same as the alignment of the receive data. Refer to tables 12.19, 12.21, 12.23, and 12.25 for details.

- Transfer request

  Receive message buffer transfer channel send a transfer request when receive data are stored to the message buffer configured as receive buffer in the message buffer of FlexRay.

  Receive message buffer transfer channel send a transfer request when receive data are been receiving to the message buffer configured as FIFO buffer in the message buffer of FlexRay.

- Addressing

  Transfer source address are fixed, transfer destination address are increment (receive) or fixed (receive FIFO).

- The details for the operation and transfer flow.

  Ch76:

  One message buffer is transferred at a transfer request. The message buffer number to be transferred is noticed. For the message buffer channel that has completed a transfer, RWF bit of A-DMAC FlexRay receive wait register (ADMAFRWR0 to 3) are set to 1. When FlexRay requires a transfer of the message buffer channel that RWF bit is set to 1, A-DMAC transfer is not accepted. When message buffer transfer at A-DMAC is completed, the flag of the corresponding channel of the FlexRay new data register in FlexRay is cleared. Figure 12.10 shows the data transfer flow.

1. In a transfer enable state (DME = 1, NMIF = 0, DE = 1), if a transfer request occurs, a transfer of 1 message buffer is performed.
2. The corresponding RWF bit to the message buffer which is performed a transfer are set to 1.
3. When NMI interrupt is occurred, or DME bit or DE bit are set to 0, a transfer is suspended (refer to 12.3.3, Transfer Suspension and Resumption). Also, during a transfer of 1 message buffer, if a transfer request is occurred to the high priority DMA channel, channels enter the transfer wait state.

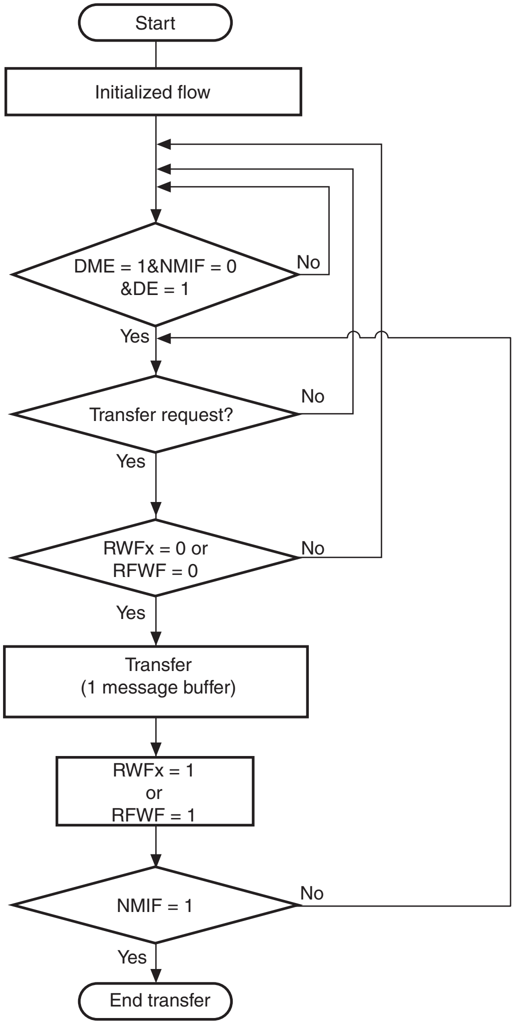
Ch77:

One message buffer is transferred per transfer request. From among the message buffers set-up for reception with FIFO, transfer proceeds for that which has the oldest received data in its FIFO. When the FlexRay module requests transfer of a message-buffer channel for which the FRWF bit is set to 1, the A-DMAC does not accept the request. When message buffer transfer at A-DMAC is completed, the flag of the corresponding channel of the FlexRay new data register in FlexRay is cleared. Figure 12.11 shows the data transfer flow.

1. In a transfer enable state (DME = 1, NMIF = 0, DE = 1), if a transfer request occurs, a transfer of 1 message buffer is performed.
2. The corresponding RWF bit to the message buffer which is performed a transfer are set to 1.
3. When NMI interrupt is occurred, or DWE bit or DE bit are set to 0, a transfer is suspended (refer to 12.3.3, Transfer Suspension and Resumption). Also, during a transfer of 1 message buffer, if a transfer request is occurred to the high priority DMA channel, channels enter the transfer wait state.

Note:

- The transfer for receive message buffer and for receive FIFO message buffer is enabled at the point of RWF and FRWF bit are cleared. Therefore, to avoid the discrepancy, refer the alias areas before clearing RWF and FRWF bits.
- The order for the data to be stored to the message buffer channel configured as receive in FlexRay does not match the order for the data to be transferred to the alias areas. This is because A-DMAC does not accept the transfer to the alias areas if RWF bit is set to 1 and the algorithm for selecting the message buffer to be transferred are not FIFO format.
- When transferring the receive data to the alias areas using the A-DMAC, do not access the message buffer for receive of FlexRay at CPU.
- When transferring the receive FIFO data to the alias areas using the A-DMAC, do not access the message buffer for receive of FlexRay at CPU. When accessed, the coherency of the message buffer transferred to the alias areas are not guaranteed.



Note: This figure has been simplified by limiting it to a case where the transfer of a single receive buffer or receive FIFO message buffer from the FlexRay module by the A-DMAC is not suspended before its completion.

Figure 12.10 Flowchart of DMA transfer

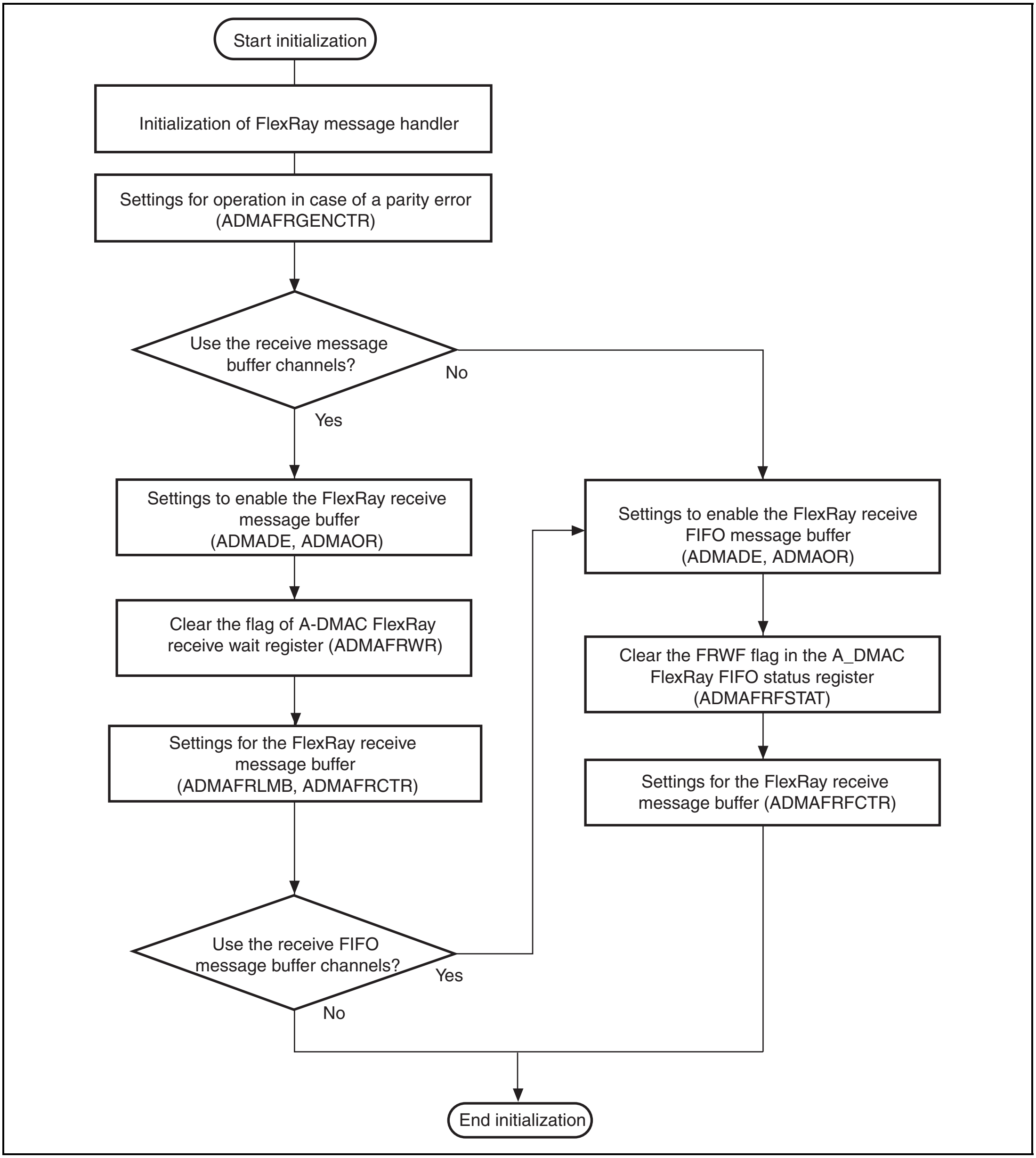

Start initialization
Initialization of FlexRay message handler
Settings for operation in case of a parity error (ADMAFRGENCTR)
Use the receive message buffer channels?
No
Yes
Settings to enable the FlexRay receive message buffer (ADMADE, ADMAOR)
Clear the flag of A-DMAC FlexRay receive wait register (ADMAFRWR)
Settings for the FlexRay receive message buffer (ADMAFRLMB, ADMAFRCTR)
Use the receive FIFO message buffer channels?
Yes
No
Settings to enable the FlexRay receive FIFO message buffer (ADMADE, ADMAOR)
Clear the FRWF flag in the A_DMAC FlexRay FIFO status register (ADMAFRFSTAT)
Settings for the FlexRay receive message buffer (ADMAFRFCTR)
End initialization

Figure 12.11 Flowchart of initialization for DMA transfer

(6) Suspension and resumption of transfer on a FlexRay channel

When suspending A-DMAC transfer operation for a channel after having enabled the transfer of a receive message buffer channel or receive FIFO message buffer channel of the FlexRay module, follow the procedure in figure 12.12. When a transfer operation for FlexRay is suspended without following this chart, the received data on the FlexRay message RAM will be disappeared.

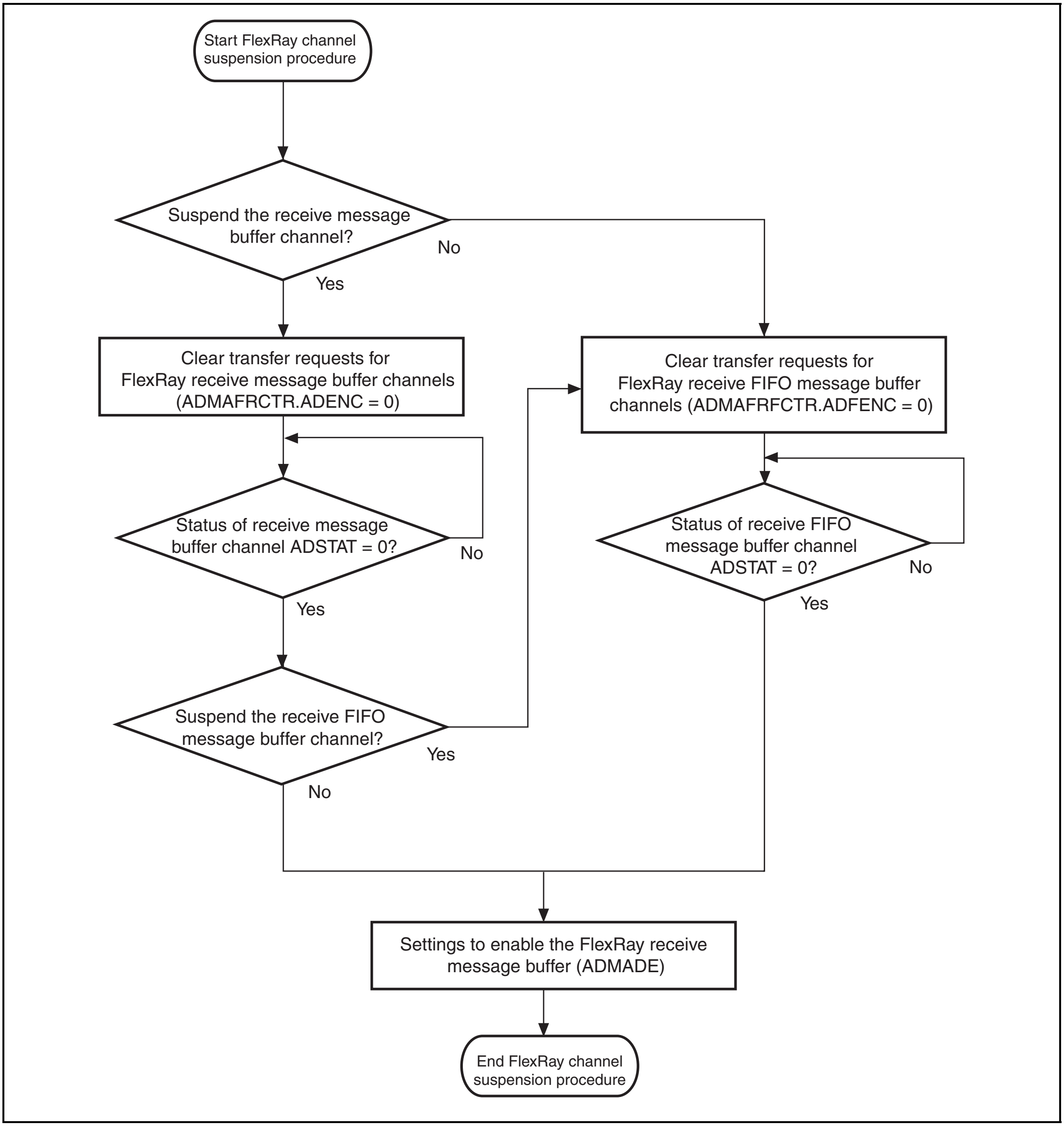

Start FlexRay channel suspension procedure
Suspend the receive message buffer channel?
No
Yes
Clear transfer requests for FlexRay receive message buffer channels (ADMAFRCTR.ADENC = 0)
Clear transfer requests for FlexRay receive FIFO message buffer channels (ADMAFRFCTR.ADFENC = 0)
Status of receive message buffer channel ADSTAT = 0?
No
Yes
Status of receive FIFO message buffer channel ADSTAT = 0?
No
Yes
Suspend the receive FIFO message buffer channel?
Yes
No
Settings to enable the FlexRay receive message buffer (ADMADE)
End FlexRay channel suspension procedure

Figure 12.12 Flow of suspension of transfer on a FlexRay channel

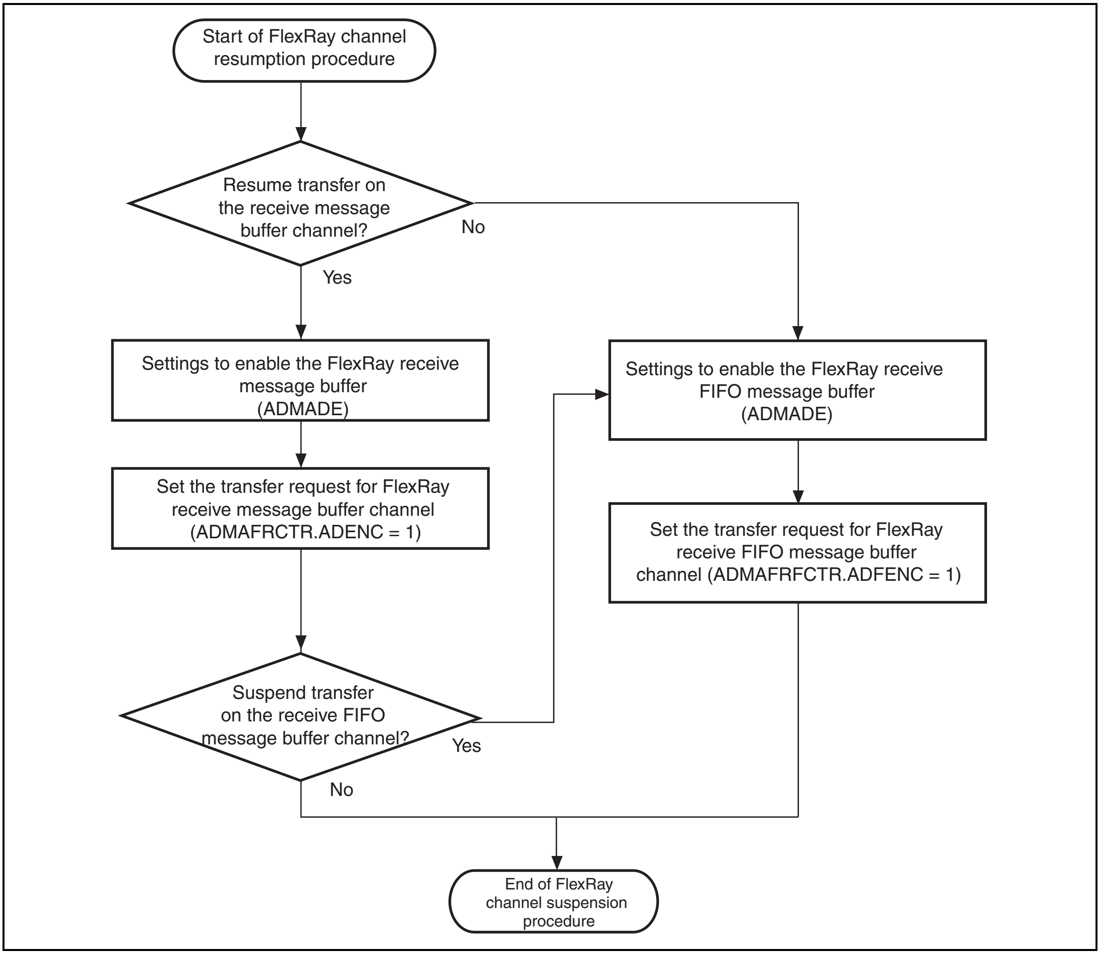

Start of FlexRay channel resumption procedure
Resume transfer on the receive message buffer channel?
No
Yes
Settings to enable the FlexRay receive message buffer (ADMADE)
Set the transfer request for FlexRay receive message buffer channel (ADMAFRCTR.ADENC = 1)
Settings to enable the FlexRay receive FIFO message buffer (ADMADE)
Set the transfer request for FlexRay receive FIFO message buffer channel (ADMAFRFCTR.ADFENC = 1)
Suspend transfer on the receive FIFO message buffer channel?
Yes
No
End of FlexRay channel suspension procedure

Figure 12.13 Flow of resumption of transfer for the FlexRay module

(7) Operation when a parity error occurs in a FlexRay channel

If the ADPEHM bit of the A-DMAC FlexRay general control register (ADMAFRGENCTR) is set to 0 and a parity error occurs during transfer of a receive buffer or receive FIFO buffer on any channel to the user RAM, the transfer operation is suspended. For recovery from the state of a transfer operation by the A-DMAC being suspending after a parity error, follow the procedure in figure 12.14.

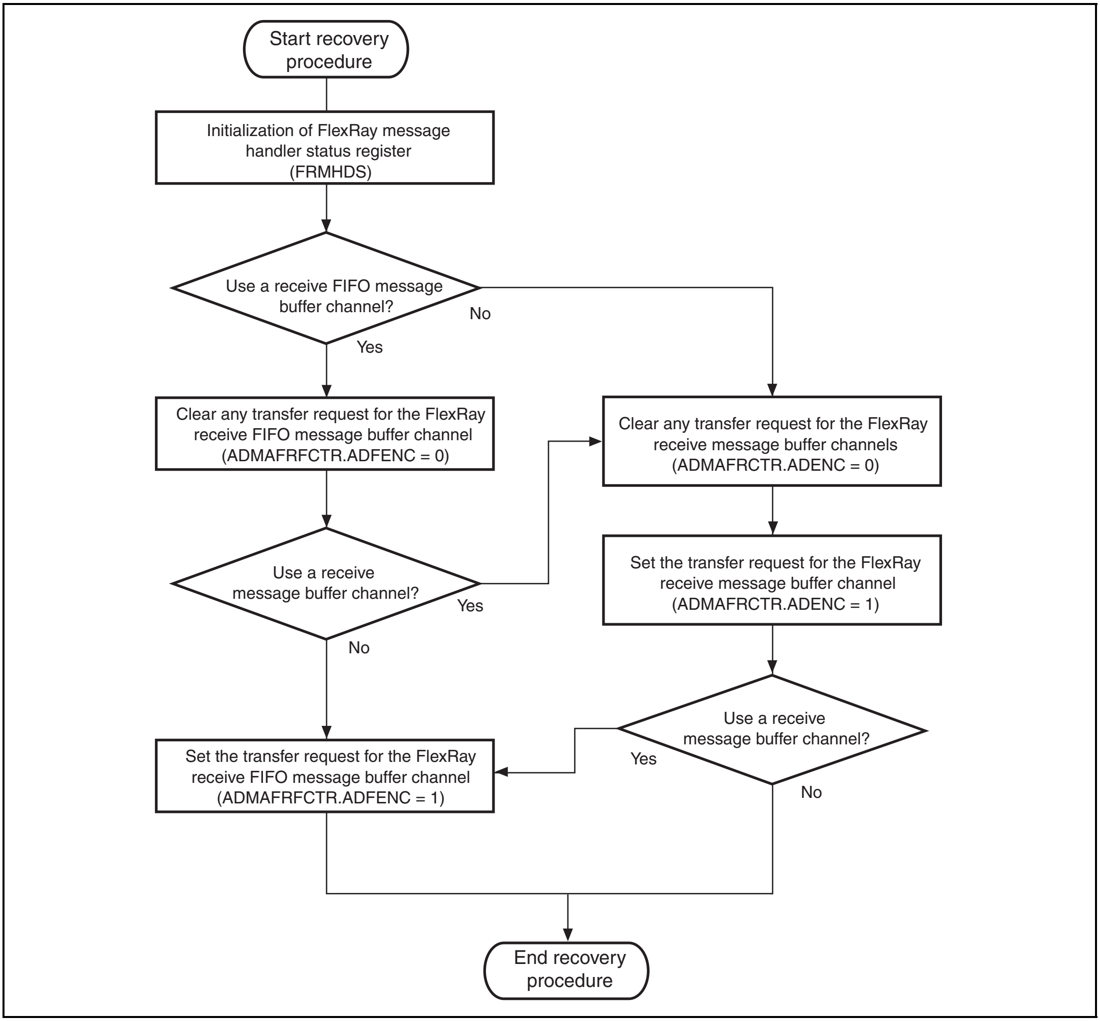

Start recovery procedure
Initialization of FlexRay message handler status register (FRMHDS)
Use a receive FIFO message buffer channel?
No
Yes
Clear any transfer request for the FlexRay receive FIFO message buffer channel (ADMAFRFCTR.ADFENC = 0)
Clear any transfer request for the FlexRay receive message buffer channels (ADMAFRCTR.ADENC = 0)
Use a receive message buffer channel?
Yes
No
Set the transfer request for the FlexRay receive message buffer channel (ADMAFRCTR.ADENC = 1)
Use a receive message buffer channel?
Yes
No
Set the transfer request for the FlexRay receive FIFO message buffer channel (ADMAFRFCTR.ADFENC = 1)
End recovery procedure

Figure 12.14 Recovery following a parity error

(8) Data in the alias area when a FlexRay parity error occurs

Refer to table 12.27 for the handling of data in the alias area in user RAM when a parity error occurs.

Table 12.27 Validity of transferred data in user RAM

ADMAFRSTAT*1							
ADMAFRFSTAT*2		ADMAFRTRSTAT*3	FRMHDS*4				
ADDPE||	ADSTAT||						
ADFDPE	ADFSTAT	ADTMB	POBF	FMBD	MFMB	FMB	Data on the alias
0	x	N	x	x	x	x	No parity error has occurred during the transfer of message buffer N. After the transfer is completed, the transfer complete flag corresponding to message buffer N is set.
1	1	N	x	x	x	x	The transfer of message buffer N is in progress. After the transfer is completed, the ADSTAT or ADFSTAT flag is cleared to 0. Also, the transfer complete flag corresponding to message buffer N is not set.
1	0	N	1	x	x	x	A parity error has occurred during the transfer of message buffer N. After the transfer is completed, the transfer complete flag corresponding to message buffer N is not set. Also, the data transferred to alias area N has a parity error.
1	0	N	x	x	1	x	A parity error has occurred during the transfer of message buffer N. After the transfer is completed, the transfer complete flag corresponding to message buffer N is not set. Also, the channel on which the parity error occurred cannot be specified, so the validity of the data transferred to alias area N cannot guaranteed.
1	0	N	0	1	0	N	When the message buffer N is transferring a parity error is occurred. After a transfer is completed, transfer complete flag corresponding to the message buffer N is not set. Also, the data transferred to N of the alias areas include a parity error.

ADMAFRSTAT*1 ADMAFRFSTAT*2		ADMAFRTRSTAT*3	FRMHDS*4				
ADDPE||	ADSTAT||						
ADFDPE	ADFSTAT	ADTMB	POBF	FMBD	MFMB	FMB	Data on the alias
1	0	N	0	1	0	M	When the message buffer M is transferring a parity error is occurred. After a transfer is completed, transfer complete flag corresponding to the message buffer N is not set. The channel where a parity error occurs is channel M, so the data transferred to N is effectiveness data which a parity error is not occurred.

[Legend]

N, M: message buffer number

x: Don' t care

Notes: 1. A-DMAC FlexRay status register
2. A-DMAC FlexRay FIFO status register
3. A-DMAC FlexRay transfer status register
4. FlexRay message handler status register

12.3.3 Transfer Suspension and Resumption

If one of the sources listed in table 12.28 occurs, data transfer is disabled (transfer is suspended). If it occurs during DMA transfer in a transfer unit, the DMA transfer in a transfer unit is performed correctly and registers such as ADMADV, ADMATE, ADMATCR, ADMAAR, ADMARVPR, ADMATVPR, ADMAFRWR, and ADMAFRFRWR are also updated correctly.

Table 12.28 Source of Transfer Suspension

Source	Channels to be Suspended
NMI interrupt occurrence (NMIF = 1 in the DMAC)	All channels
Clearing the DME bit to 0	
Clearing the DE bit to 0	Corresponding channel

If one of the sources listed in table 12.28 is cancelled, A-DMAC enters the transfer enable state (transfer request wait state). A-DMAC channels for the RSPI and SCI refer to the current values for the next transfer request, and perform transfer (transfer is resumed).

If the transfer is suspended during MBx transfer, the A-DMAC channel for RCAN operation differs depending on the source of the transfer suspension, as follows.

An NMI interrupt is cancelled: The next data of MBx being transferred is transferred (transfer is resumed).

The DME (DE) bit is set to 1: Data is transferred from the start of MBx. (In the A-DMA channel for RCAN reception, MBx is informed synchronously with the next transfer request. In the A-DMAC channel for RCAN transmission, MBx has the highest priority.)

Furthermore, when the transfer of a message buffer on a FlexRay channel is suspended before it is complete, operation varies with the reason for the suspension.

Release from NMI interrupt: Transfer the next data from the message buffer (resumption of transfer).
Set the DME (DE) bit to 1: Unless the procedure shown in section 12.3.2 (6), Suspension and resumption of transfer on a FlexRay channel, operation after clearing the DME (DE) bit to 0 is not guaranteed.

12.3.4 Bus Operation in Data Transfer by A-DMAC

- Addressing mode

  Performs the operation corresponding to the dual address mode in DMAC. Accordingly, in the read cycle, the transfer source is accessed; while in the write cycle, the transfer destination is accessed. The transfer size is fixed in each channel.

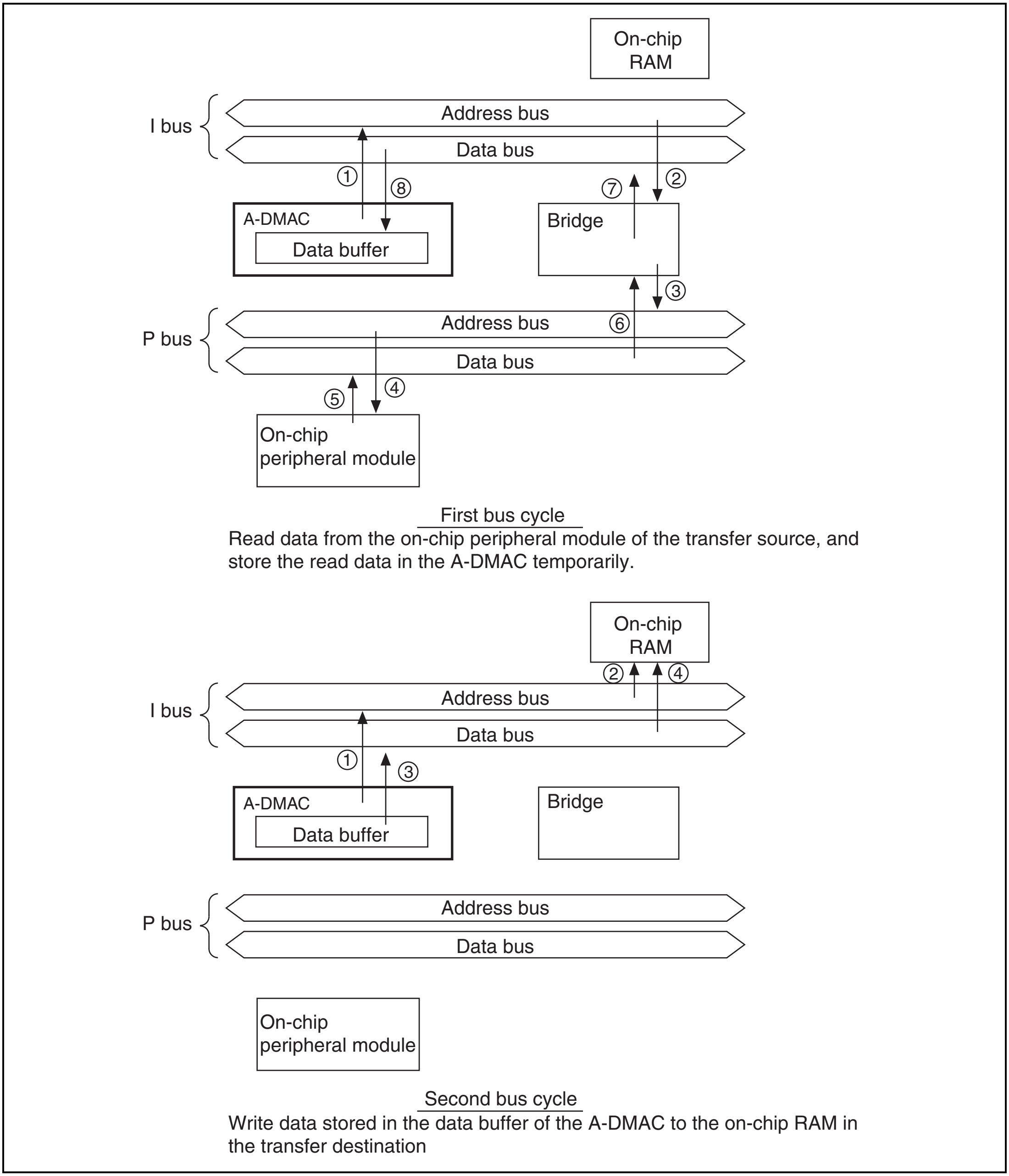
- Bus cycle

  The A-DMAC requests the bus mastership of I bus if it receives a transfer request. Upon acquiring the I-bus mastership, the A-DMAC starts one unit of DMA transfer consisting of two bus cycles of read and write. In one unit of DMA transfer, the A-DMAC does not release the I-bus mastership. Figure 12.15 shows the data for data transfer performed from the on-chip peripheral module to the on-chip RAM.

Note: In the A-DMAC channels for ATU-III (timers A and F) and the RSPI, one unit of DMA transfer does not include two bus cycles.

- Bus mode

  The A-DMAC performs data transfer in cycle steal mode. The A-DMAC releases the I-bus mastership after one unit of DMA transfer even if another A-DMAC channel waits for DMA transfer (for details, refer to section 12.3.5, Channel Priorities).



First bus cycle
Read data from the on-chip peripheral module of the transfer source, and store the read data in the A-DMAC temporarily.

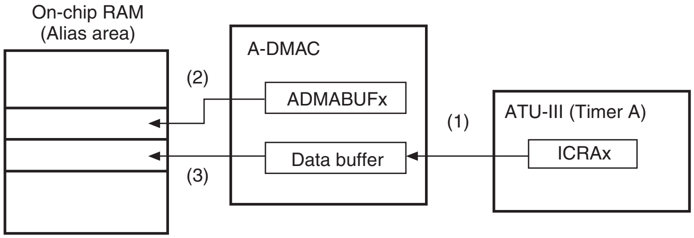
Second bus cycle
Write data stored in the data buffer of the A-DMAC to the on-chip RAM in the transfer destination

Figure 12.15 Data Flow

(1) Supplementary Description for A-DMAC Channels Used for ATU-III (timer A)

One unit of DMA transfer consists of three bus cycles of read, write, and write. Figure 12.16 shows the data flow of the A-DMAC channel for timer A.

Note: Because the ADMABUF value, which is transferred to the alias area after DE is changed from 0 to 1, is not the input captured value, the ADMABUF value has no meaning. Accordingly, this register value must be ignored.



[Data flow]

(1) Read the input capture register.

(2) Write the ADMABUF register value to the upper 32 bits of the alias area.

(3) Write the input capture register value to the lower 32 bits of the alias area.

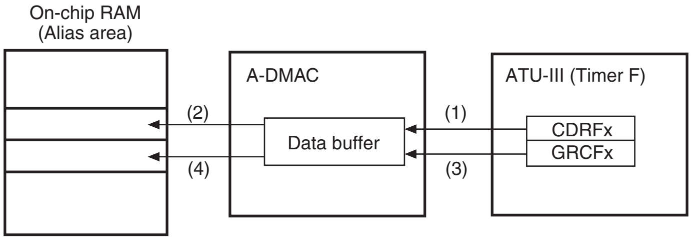
Note: The A-DMAC does not release the I-bus mastership during one DMA transfer.

Figure 12.16 Data Flow in Channels for Timer A

(2) Supplementary Description for A-DMAC Channels Used for ATU-III (timer F)

When the MODE bit in ADMAMODE is set to 1, one unit of DMA transfer consists of four bus cycles of read, write, read, and write. Figure 12.17 shows the data flow of A-DMAC channel for timer F.

When the MODE bit in ADMAMODE is cleared to 0, one unit of DMA transfer consists of two bus cycles of read and write. For details, refer to section 12.3.4, Bus Operation in Data Transfer by A-DMAC.



[Data flow]

(1) Read CDRFx.

(2) Write data to the upper 32 bits of the alias area.

(3) Read the GRCFx register (only when MODE = 1).

(4) Write data to the lower 32 bits of the alias area (only when MODE = 1).

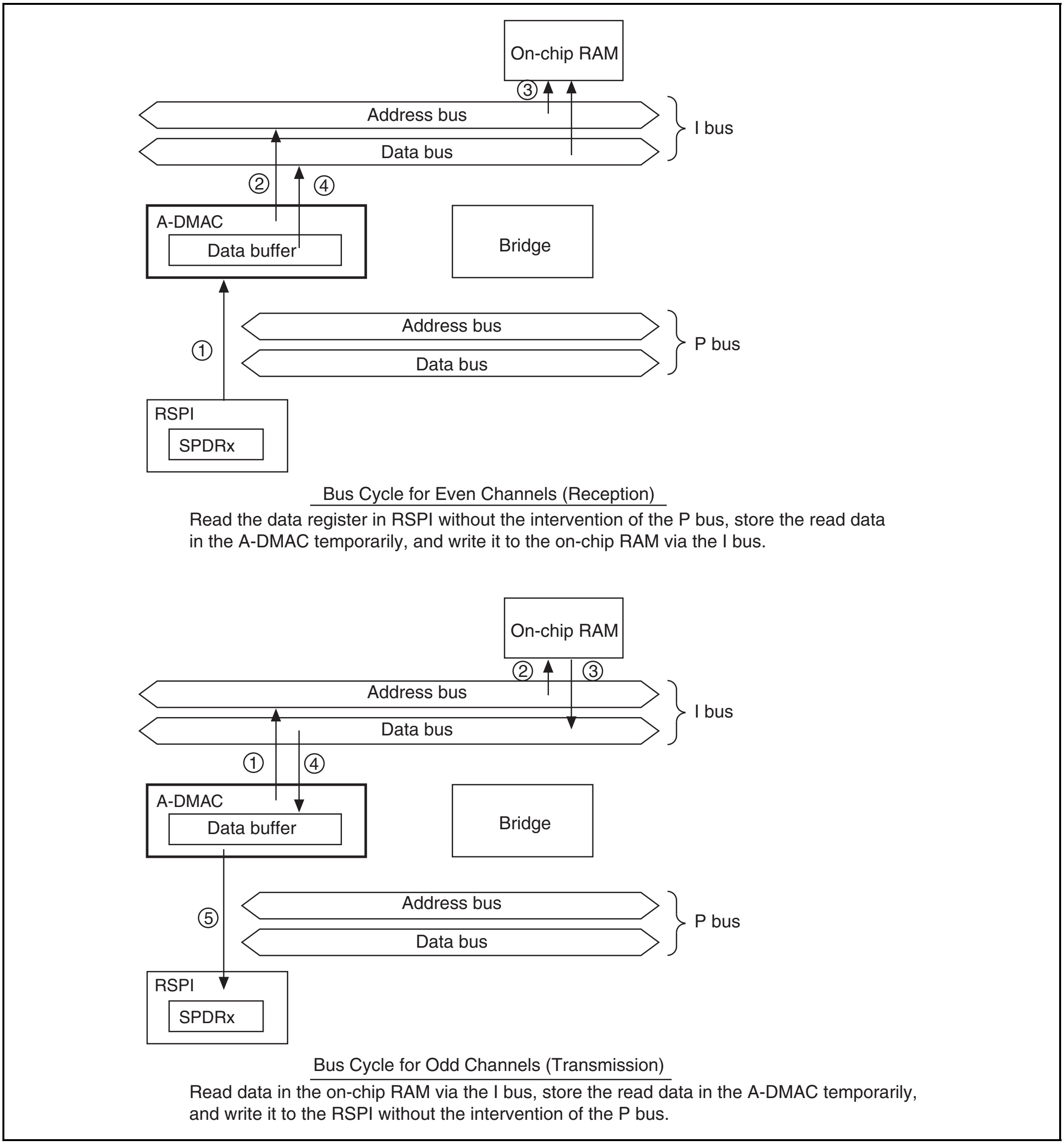
Note: The A-DMAC does not release the I-bus mastership during one DMA transfer.

Figure 12.17 Data Flow in Channels for Timer F

(3) Supplementary Description for A-DMAC Channels Used for RSPI

Because the RSPI is connected to the A-DMAC via the specific bus, the A-DMAC can access the RSPI data register (SPDR) without the intervention of the P bus. Accordingly, one unit of DMA transfer in the transmission channel can be performed in one bus cycle of read from the I bus (alias area read) and that in the reception channel can be performed in one bus cycle of write to the I bus (alias area write). As a result, high-speed data transfer between the RSPI and on-chip RAM can be performed.

Figure 12.18 shows the data flow of A-DMAC channel for the RSPI.



Bus Cycle for Even Channels (Reception)

Read the data register in RSPI without the intervention of the P bus, store the read data in the A-DMAC temporarily, and write it to the on-chip RAM via the I bus.

Bus Cycle for Odd Channels (Transmission)

Read data in the on-chip RAM via the I bus, store the read data in the A-DMAC temporarily, and write it to the RSPI without the intervention of the P bus.

Figure 12.18 Data Flow in Channels for RSPI

12.3.5 Channel Priorities

If multiple transfer requests to multiple channels occur simultaneously, the transfer request is accepted according to the channel priority. The channel priority is fixed (channel 0 > channel 1 > … > channel 74 > channel 75 > channel 76> channel 77).

If a transfer request for a channel of higher priority occurs during data transfer in the channel for the RCAN, the data transfer in channel for the RCAN is suspended and the data transfer of the high-priority channel is performed. If a transfer request for a RCAN reception channel occurs during data transfer in the RCAN transmission channel, the current data transfer in the RCAN transmission channel is suspended and enters the wait state until data transfer in the RCAN reception channel has been completed.

Figure 12.19 shows the A-DMAC operation when a transfer request for channels 0 and 72 occurs simultaneously and when a transfer request for channel 1 occurs during channel 72 operation.

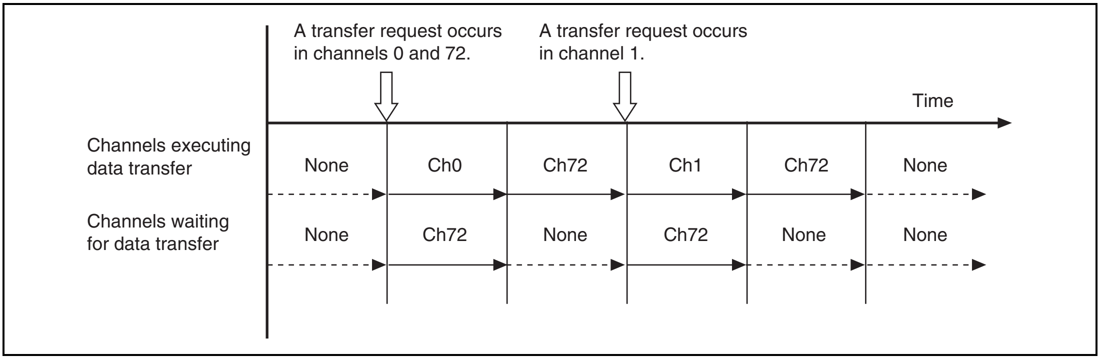

A transfer request occurs in channels 0 and 72.
A transfer request occurs in channel 1.
Time
Channels executing data transfer
None
Ch0
Ch72
Ch1
Ch72
None
Channels waiting for data transfer
None
Ch72
None
Ch72
None
None

Figure 12.19 Channel Priorities

12.3.6 Transfer Enable/Disable Conditions and Interrupt Requests

Table 12.29 lists the transfer enable and disable conditions for each channel.

The A-DMAC uses an interrupt request signal from the peripheral module as a transfer activation source.

The A-DMAC requests an interrupt under the following two conditions.

(1) Transfer End Interrupt by the A-DMAC (other than channels 0, 1, 72, 73, 76, and 77)

In the transfer enable state, an interrupt request signal from the peripheral module is used as an A-DMAC transfer activation source and is not informed to the INTC. If the A-DMAC transfer using the interrupt source has been completed, an A-DMAC transfer end is notified to the INCT (transfer end interrupt) at the timing when the DV or TE bit is set to 1. The transfer end interrupt is cleared by clearing the TE or DV bit to 0.

A transfer end interrupt can be masked by clearing the IE bit to 0.

(2) Passing through of Transfer Request Interrupts from Peripheral Modules (other than channels 72, 73, 74, 75, 76, and 77)

Interrupt request signals from peripheral modules are masked by the A-DMAC in the 'transfer-disabled state (masked)' shown in table 12.29. On the other hand, interrupt request signals from peripheral modules are directly informed to the INTC in the 'transfer-disabled state (pass-through)' shown in table 12.29. The transfer requests that have been passed through cannot be masked even if the IE bit is cleared to 0.

Notes:

- For details on the connections between the A-DMAC and INTC and the transfer request masking by the A-DMAC and DMAC, refer to section 8, Interrupt Controller (INTC).
- The INTC handles interrupt requests from each A-DMAC channel as different interrupt sources.

  The INTC assigns an A-DMAC transfer end interrupt and transfer request interrupt (pass-through) for the same interrupt from the peripheral module to the same interrupt vector.
- Transfer request interrupts from channels 8, 12, 16, 20 and 24, which correspond to GRCx0 of timer C, are informed to the INTC via the DMAC. Transfer request interrupts from other channels are informed to the INTC directly.
- In the transfer-disabled (masked) state, a DMA transfer request is masked by the A-DMAC. In this case, data transfer is not performed and the transfer request interrupt is not informed to the INTC.
- If the transfer-suspended state is entered (for details, refer to section 12.3.3, Transfer Suspension and Resumption) while the transfer end interrupt is not yet cleared, the interrupt request is retained.

Table 12.29 (1) Transfer Enable and Disable Conditions for Channels Used for ATU-III (Timer G) and ADC

NMI	DME	TCR	Transfer Request
1	×	×	Transfer disabled (masked)
0	0	×	Transfer disabled (pass-through)
	1	TCR = 0	Transfer disabled (pass-through)
		TCR != 0	Transfer enabled

[Legend]

×: Don't care

Table 12.29 (2) Transfer Enable and Disable Conditions for Channels Used for ATU-III (Timers A, C, and F)

NMI	DME	DE	DV	Transfer Request
1	×	×	×	Transfer disabled (masked)
0	0	×	×	Transfer disabled (pass-through)
	1	0	×	Transfer disabled (pass-through)
		1	×	Transfer enabled

[Legend]

×: Don't care

Table 12.29 (3) Transfer Enable and Disable Conditions for Channels Used for RSPI and SCI

NMI	DME	TCR	TE	Transfer Request
1	×	×	×	Transfer disabled (masked)
0	0	×	×	Transfer disabled (pass-through)
	1	TCR = 0	0	Transfer disabled (pass-through)
			1	Transfer disabled (masked)
		TCR != 0	0	Transfer enabled
			1	Transfer disabled (masked)

[Legend]

×: Don't care

Table 12.29 (4) Transfer Enable and Disable Conditions for Channels Used for RCAN (Reception)

NMI	DME	DE	Transfer Request
1	×	×	Transfer disabled
0	0	×	Transfer disabled
	1	0	Transfer disabled
		1	Transfer enabled

[Legend]

×: Don't care

Table 12.29 (5) Transfer Enable and Disable Conditions for Channels Used for RCAN (Transmission)

NMI	DME	DE	TE	Transfer Request
1	×	×	×	Transfer disabled
0	0	×	×	Transfer disabled
	1	0	×	Transfer disabled
		1	0	Transfer enabled
			1	Transfer disabled

[Legend]

×: Don't care

Table 12.29 (6) State of Enabling or Disabling Channels for Use in Transferring FlexRay Receive and Receive FIFO Message Buffers

NMI	DME	DE	Transfer Request
1	×	×	Transfer disabled
0	0	×	Transfer disabled
	1	0	Transfer disabled
		1	Transfer enabled

[Legend]

×: Don't care

12.4 Usage Note

In ADMADV, ADMATE, and ADMARVPR, when the flag is read by the CPU, even though 0 has been read, the flag may be set to 1 and therefore 1 may be read internally.

In this case, if 0 (in ADMADV or ADMATE) or 1 (in ADMARVPR) is written to the corresponding bit, the flag will be cleared even if 1 has not been read by the CPU.

To avoid the malfunction, the clear conditions must be observed, described in the notes of ADMADV, ADMATE, and ADMARVPR.

Since the specifications of the flag bits in these registers differ from the ones of the flags in other modules, the clearing operations of the flags by the CPU must be executed carefully.

Section 13 Advanced Timer Unit III (ATU-III)

ATU-III consists of nine timer blocks (timer A to timer J), prescalers, and a controller. The timer blocks have different functions and each can operate independently; timer blocks can also be linked via the clock bus. Each timer block consists of one or more timer subblocks and each subblock has one or more channels. (See table 13.1).

13.1 Features

- Processing of up to 118 pulse input/output signals
- 158 interrupt sources can be generated. This enables to activate the direct memory access controller (DMAC), automotive direct memory access controller (A-DMAC), and interrupt processing by the CPU.
- 22 pulse output dedicated for A/D (16 for timer D and 6 for timer G)
- On-chip 4-channel prescaler provided, wvhich generates four types of clocks by dividing on-chip peripheral clock (Pϕ) by 1/1 to 1/1024
- Each channel for a timer can select a count source from among four divided clocks generated by prescaler, two external clocks, and angle clock generated by timer B.

(1) Timer A

Timer A has a 32-bit free-run counter and six 32-bit input capture registers. Features are shown below.

- Detection by rising edges, falling edges, or both edges
- A-DMAC activation at capture timing
- Noise canceling function for each external pin with maximum length of 1.64 ms
- Capture interrupt and counter overflow interrupt are available
- Settings for noise cancellation mode can be made in channel units.

(2) Timer B

Timer B consists of three subblocks: an edge-interval measuring block, frequency-multiplied clock generator, and frequency-multiplied clock signal corrector.

1. The edge-interval measuring block is provided with a 32-bit input edge-interval measuring timer, output compare and input capture registers (three registers), 8-bit event counter, and output compare register. This provides the following operations:

- Capture by edges of external event input (rising edge, falling edge, or both edges are selectable)
- Capture by event compare match of external event input
- Capture interrupt and compare match interrupt (edge-interval compare match, event compare match)

2. The frequency-multiplied clock generator is provided with 24-bit reloadable counter, reload register, 20-bit multiplied clock counter, and output compare register. This provides the following operations:

- Reloadable counting of values captured by edge-interval measuring block with arbitrary number (1 to 4095)
- Internal clock generated by the underflow of reloadable counter can be used as input of 20-bit multiplied clock counter
- Compare match interrupt generation.

3. Frequency-multiplied clock signal corrector is provided with 20-bit correcting event counter, 20-bit correcting multiplied clock counter, multiplied-and-corrected clock generating counter, and correcting counter clearing register. This provides the following operations:

- Frequency-multiplied correcting clock that serves as the count source for other timers can be generated based on the read count in frequency-multiplied clock generator.
- Free-run counter for timer D can be cleared by multiplied-and-corrected clock generating counter and correcting counter clearing register

(3) Timer C

Timer C consists of five subblocks that have the same functions. Each subblock consists of four channels. Each subblock is provided with one 24-bit free running counter and four 24-bit general registers. This provides the following operations:

- Input capture or output compare is selectable
- Detection edge for input capture is selectable from among rising edge, falling edge, or both edges.
- Each input capture trigger input has a function that cancels noise with maximum length of 1.64 ms.
- 1, 0, or toggle can be output by a compare match.
- Three PWM waveforms can be output for each subblock in PWM mode.
- Input capture/compare match interrupt and overflow interrupt can be generated. 28 input capture/compare match interrupts activate A-DMAC and five interrupts activate DMAC.

(4) Timer D

Timer D consists of four subblocks that have the same function. Each subblock consists of four channels. Each subblock is provided with two 24-bit free run counter, offset base register, and four channels. Each channel is provided with four 24-bit output compare registers, four general registers, and four 24-bit down counter for outputting one-shot pulse. This provides the following operations:

- Enables to start downcounter by software. One-shot pulse can also be generated.
- Compare match between compare match register and general register can be used as the start trigger for downcounter. One-shot pulse with offset can also be generated.
- Compare match in general register can stop downcounter and forcibly cut off one-shot pulse output.
- Compare match between compare match register and general register can be output.
- Compare match in compare match register can be used as a trigger and the count number can be captured by general register.
- Free run value can be captured triggered by timer A
- Provided with counter clearing function from timer B
- Generation of sixteen compare match interrpts, eight counter overflow interrupts, and sixteen underflow interrupts. The sixteen undeflow interrupts correspond to DMAC activation.
- Output pulses that show interrupts of compare match A or compare match B for A/D activation. (eight for each)
- Output waveform can be inverted.

(5) Timer E

Timer E consists of seven subblocks that have the same function. Each subblock consists of four channels. Each channel is provided with 16-bit free running counter, duty cycle setting register, cycle setting register, duty cycle reload register, and cycle reload register. This provides the following operations:

- PWM output with programmable cycle time and duty cycle ranging from 0 to 100%
- Switchable between on-state and off-state duty modes
- Values in duty cycle reload register/cycle reload register can be transferred to duty cycle setting register/cycle setting register at every cycle.
- Writng H'0000 to counter can forcibly end PWM cycle and start new PWM cycle.
- Periodic output of interrupt requests (up to twenty eight; of these, six can be used to activate the DMAC).

(6) Timer F

Timer F consists of 28 subblocks. Each subblock is provided with two 24-bit counters, a 16-bit counter, three 24-bit general registers (for subblocks 12 to 15 only, two for other subblocks), and a 16-bit general register. This provides the following operations:

- Noise canceling function for each external pin with maximum length of 1.64 ms
- Seven operation modes: edge counting in a specified period, valid edge interval counting, measurement of time during high/low input levels, measurement of PWM input waveform timing, rotation speed/pulse measurement, up/down event count, and four-time multiplication event count.
- Activates A-DMAC by input capture interrupt
- Overflow interrupt generation
- Settings for noise cancellation mode can be made in channel units.

(7) Timer G

Timer G consists of six subblocks that have the same function. Each channel is provided with a 16-bit free-run counter and output compare register. This provides the following operations:

- Outputs event that is triggered by compare match. This output can be used as a trigger for AD activation/interrupt.
- Activates DMAC by compare match interrupt

(8) Timer H

Timer H consists of a pair of 16-bit countr and 16-bit compare match register, and a 32-bit counter. This provides the following operations:

- Measurement of time ranging from 1 to 2^{26} times of internal peripheral clock (Pϕ) using 16-bit counter and 16-bit compare match register. Can be output as compare match interrupt.
- Equipped with a 32-bit counter to count the compare match occurrence.

(9) Timer J

Timer J consists of two subblocks that have the same function. Each channel is provided with a 16-bit counter, output compare register, and nine-stage FIFO register. This provides the following operations:

- Detection by rising edge, falling edge or both edges
- Noise canceling function for each external pin with maximum length of 1.64 ms

- Capture the counter value in FIFO register (edge input interval) when detecting edges on external input pin.
- Activates DMAC at the timing when FIFO is full.
- Controls FIFO's effective capture time using compare match register
- FIFO full interrupt, counter overflow, and FIFO overflow interrupt can be generated.

Table 13.1 Block configuration for ATU-III

Modules	Blocks	Subblock		Remarks
ATU-III				
	Common controller unit			
	Prescaller			Total number of channels: 4
		Channel 0		
		. . .		
		Channel 3		
	Timer A			Total number of channels: 6
		Channel 0		
		. . .		
		Channel 5		
	Timer B			Total number of channels: 1
	Timer C			Total number of channels: 20 Number of subblock: 5 Number of channel/subblock: 4
		Timer C0		
			Channel 0	
			. . .	
			Channel 3	
		Timer C1 to Timer C4		
	Timer D			Total number of channels: 16 Number of subblocks: 4 Number of channels/subblock: 4
		Timer D0		
			Channel 0	
			. . .	
			Channel 3	
		Timer D1 to Timer D3		

Table 13.1 Block configuration for ATU-III (cont)

	Timer E			Total number of channels: 28 Number of subblocks: 7 Number of channels/subblock: 4
		Timer E0		
			Channel 0	
			. . .	
			Channel 3	
		Timer E1 to Timer E6		
	Timer F			Total number of channels: 28 Number of subblocks: 28 Number of channels/subblock: 1
		Timer F0		
		. . .		
		Timer F27		
	Timer G			Total number of channels: 6 Number of subblocks: 6 Number of channels/subblock: 1
		Timer G0		
		. . .		
		Timer G5		
	Timer H			Total number of channels: 1
	Timer J			Total number of channels: 2 Number of subblocks: 2 Number of channels/subblock: 1
		Timer J0		
		Timer J1		

13.2 Register Addresses

Addresses of the ATU-III registers are shown below. To access the registers, the following procedure should be followed.

- When writing to reserved bits, the value written must be 0.
- Registers which have more than 16 bits must be read from and written to in 32 bit units. These registers cannot be accessed in 16- or 8-bit units.

Table 13.2 Common Controller Registers

Address	31 24	23 16	15 8	7 0
H'FFFF F000	ATUENR		CBCNT	NCMR
H'FFFF F004 : H'FFFF F0FC	Reserved			

Table 13.3 Prescaler Registers

Address	31 24	23 16	15 8	7 0
H'FFFF F100	PSCR0		PSCR1	
H'FFFF F104	PSCR2		PSCR3	
H'FFFF F108 : H'FFFF F1FC	Reserved			

Table 13.4 Timer A Registers

Address	31 24	23 16	15 8	7 0
H'FFFF F200	Reserved		TCRA	Reserved
H'FFFF F204	TIOR1A		TIOR2A	
H'FFFF F208	TSRA	TIERA	NCMCR1A	Reserved
H'FFFF F20C	Reserved			
H'FFFF F210	NCNTA0	NCRA0	NCNTA1	NCRA1
H'FFFF F214	NCNTA2	NCRA2	NCNTA3	NCRA3
H'FFFF F218	NCNTA4	NCRA4	NCNTA5	NCRA5
H'FFFF F21C	Reserved			
H'FFFF F220	TCNTA			
H'FFFF F224	Reserved			
H'FFFF F228	ICRA0			
H'FFFF F22C	ICRA1			
H'FFFF F230	ICRA2			
H'FFFF F234	ICRA3			
H'FFFF F238	ICRA4			
H'FFFF F23C	ICRA5			
H'FFFF F240 : H'FFFF F2FC	Reserved			

Table 13.5 Timer B Registers

Address	31 24	23 16	15 8	7 0
H'FFFF F300	Reserved			
H'FFFF F304	TCRB	TIORB	TSRB	TIERB
H'FFFF F308 : H'FFFF F30C	Reserved			
H'FFFF F310	TCNTB0			
H'FFFF F314	ICRB0			
H'FFFF F318	OCRB0			
H'FFFF F31C	TCNTB1	OCRB1	Reserved	
H'FFFF F320	ICRB1			
H'FFFF F324	ICRB2			
H'FFFF F328	Reserved			
H'FFFF F32C				
H'FFFF F330	LDB			
H'FFFF F334	RLDB			
H'FFFF F338	PIMR		Reserved	
H'FFFF F33C	TCNTB2			
H'FFFF F340	TCNTB6			
H'FFFF F344	OCRB6			
H'FFFF F348	OCRB7			
H'FFFF F34C	Reserved			
H'FFFF F350	TCNTB3			
H'FFFF F354	TCNTB4			
H'FFFF F358	TCNTB5			
H'FFFF F35C	TCCLRB			
H'FFFF F360 : H'FFFF F3FC	Reserved			

Table 13.6 Timer C Registers

Address	31 24	23 16	15 8	7 0
H'FFFF F400	TSTRC	Reserved	NCCRC0	NCCRC1
H'FFFF F404	NCCRC2	NCCRC3	NCCRC4	Reserved
H'FFFF F408	Reserved			
H'FFFF F40C				
H'FFFF F410	NCNTC00	NCNTC01	NCNTC02	NCNTC03
H'FFFF F414	NCRC00	NCRC01	NCRC02	NCRC03
H'FFFF F418	NCNTC10	NCNTC11	NCNTC12	NCNTC13
H'FFFF F41C	NCRC10	NCRC11	NCRC12	NCRC13
H'FFFF F420	NCNTC20	NCNTC21	NCNTC22	NCNTC23
H'FFFF F424	NCRC20	NCRC21	NCRC22	NCRC23
H'FFFF F428	NCNTC30	NCNTC31	NCNTC32	NCNTC33
H'FFFF F42C	NCRC30	NCRC31	NCRC32	NCRC33
H'FFFF F430	NCNTC40	NCNTC41	NCNTC42	NCNTC43
H'FFFF F434	NCRC40	NCRC41	NCRC42	NCRC43
H'FFFF F438	Reserved			
H'FFFF F43C				
H'FFFF F440	TCRC0	TIERC0	TIORC0	
H'FFFF F444	TSRC0	Reserved		
H'FFFF F448	GRC00			
H'FFFF F44C	GRC01			
H'FFFF F450	GRC02			
H'FFFF F454	GRC03			
H'FFFF F458	TCNTC0			
H'FFFF F45C	Reserved			
H'FFFF F460	TCRC1	TIERC1	TIORC1	
H'FFFF F464	TSRC1	Reserved		
H'FFFF F468	GRC10			
H'FFFF F46C	GRC11			
H'FFFF F470	GRC12			
H'FFFF F474	GRC13			
H'FFFF F478	TCNTC1			

Address	31 24	23 16	15 8	7 0
H'FFFF F47C	Reserved			
H'FFFF F480	TCRC2	TIERC2	TIORC2	
H'FFFF F484	TSRC2	Reserved		
H'FFFF F488	GRC20			
H'FFFF F48C	GRC21			
H'FFFF F490	GRC22			
H'FFFF F494	GRC23			
H'FFFF F498	TCNTC2			
H'FFFF F49C	Reserved			
H'FFFF F4A0	TCRC3	TIERC3	TIORC3	
H'FFFF F4A4	TSRC3	Reserved		
H'FFFF F4A8	GRC30			
H'FFFF F4AC	GRC31			
H'FFFF F4B0	GRC32			
H'FFFF F4B4	GRC33			
H'FFFF F4B8	TCNTC3			
H'FFFF F4BC	Reserved			
H'FFFF F4C0	TCRC4	TIERC4	TIORC4	
H'FFFF F4C4	TSRC4	Reserved		
H'FFFF F4C8	GRC40			
H'FFFF F4CC	GRC41			
H'FFFF F4D0	GRC42			
H'FFFF F4D4	GRC43			
H'FFFF F4D8	TCNTC4			
H'FFFF F4DC	Reserved			
H'FFFF F4E0 : H'FFFF F4FC	Reserved			

Table 13.7 Timer D Registers

Address	31 24	23 16	15 8	7 0
H'FFFF F500	TSTRD	Reserved		
H'FFFF F504 : H'FFFF F51F	Reserved			
H'FFFF F520	TCNT1D0			
H'FFFF F524	TCNT2D0			
H'FFFF F528	OSBRD0			
H'FFFF F52C	TCRD0		TOCRD0	CMPOD0
H'FFFF F530	TCNT1D1			
H'FFFF F534	TCNT2D1			
H'FFFF F538	OSBRD1			
H'FFFF F53C	TCRD1		TOCRD1	CMPOD1
H'FFFF F540	TCNT1D2			
H'FFFF F544	TCNT2D2			
H'FFFF F548	OSBRD2			
H'FFFF F54C	TCRD2		TOCRD2	Reserved
H'FFFF F550	TCNT1D3			
H'FFFF F554	TCNT2D3			
H'FFFF F558	OSBRD3			
H'FFFF F55C	TCRD3		TOCRD3	Reserved
H'FFFF F560 : H'FFFF F57C	Reserved			
H'FFFF F580	TIOR1D0		TIOR2D0	
H'FFFF F584	Reserved	DSTRD0	Reserved	DSRD0
H'FFFF F588	DCRD0		Reserved	
H'FFFF F58C	TSRD0		TIERD0	
H'FFFF F590	OCRD00			
H'FFFF F594	OCRD01			
H'FFFF F598	OCRD02			

Address	31 24	23 16	15 8	7 0
H'FFFF F59C	OCRD03			
H'FFFF F5A0	GRD00			
H'FFFF F5A4	GRD01			
H'FFFF F5A8	GRD02			
H'FFFF F5AC	GRD03			
H'FFFF F5B0	DCNTD00			
H'FFFF F5B4	DCNTD01			
H'FFFF F5B8	DCNTD02			
H'FFFF F5BC	DCNTD03			
H'FFFF F5C0	TIOR1D1		TIOR2D1	
H'FFFF F5C4	Reserved	DSTRD1	Reserved	DSRD1
H'FFFF F5C8	DCRD1		Reserved	
H'FFFF F5CC	TSRD1		TIERD1	
H'FFFF F5D0	OCRD10			
H'FFFF F5D4	OCRD11			
H'FFFF F5D8	OCRD12			
H'FFFF F5DC	OCRD13			
H'FFFF F5E0	GRD10			
H'FFFF F5E4	GRD11			
H'FFFF F5E8	GRD12			
H'FFFF F5EC	GRD13			
H'FFFF F5F0	DCNTD10			
H'FFFF F5F4	DCNTD11			
H'FFFF F5F8	DCNTD12			
H'FFFF F5FC	DCNTD13			
H'FFFF F600	TIOR1D2		TIOR2D2	
H'FFFF F604	Reserved	DSTRD2	Reserved	DSRD2
H'FFFF F608	DCRD2		Reserved	
H'FFFF F60C	TSRD2		TIERD2	
H'FFFF F610	OCRD20			
H'FFFF F614	OCRD21			
H'FFFF F618	OCRD22			

Address	**31 24**	**23 16**	**15 8**	**7 0**
H'FFFF F61C	OCRD23			
H'FFFF F620	GRD20			
H'FFFF F624	GRD21			
H'FFFF F628	GRD22			
H'FFFF F62C	GRD23			
H'FFFF F630	DCNTD20			
H'FFFF F634	DCNTD21			
H'FFFF F638	DCNTD22			
H'FFFF F63C	DCNTD23			
H'FFFF F640	TIOR1D3		TIOR2D3	
H'FFFF F644	Reserved	DSTRD3	Reserved	DSRD3
H'FFFF F648	DCRD3		Reserved	
H'FFFF F64C	TSRD3		TIERD3	
H'FFFF F650	OCRD30			
H'FFFF F654	OCRD31			
H'FFFF F658	OCRD32			
H'FFFF F65C	OCRD33			
H'FFFF F660	GRD30			
H'FFFF F664	GRD31			
H'FFFF F668	GRD32			
H'FFFF F66C	GRD33			
H'FFFF F670	DCNTD30			
H'FFFF F674	DCNTD31			
H'FFFF F678	DCNTD32			
H'FFFF F67C	DCNTD33			
H'FFFF F680 : H'FFFF F6FF	Reserved			

Table 13.8 Timer E Registers

Address	31 24	23 16	15 8	7 0
H'FFFF F700	TSTRE	Reserved		
H'FFFF F704 : H'FFFF F7FC	Reserved			
H'FFFF F800	TCRE0	TOCRE0	TIERE0	RLDCRE0
H'FFFF F804	TSRE0	Reserved		
H'FFFF F808	PSCRE0	Reserved		
H'FFFF F80C	SSTRE0	Reserved		
H'FFFF F810	CYLRE00		CYLRE01	
H'FFFF F814	CYLRE02		CYLRE03	
H'FFFF F818	DTRE00		DTRE01	
H'FFFF F81C	DTRE02		DTRE03	
H'FFFF F820	CRLDE00		CRLDE01	
H'FFFF F824	CRLDE02		CRLDE03	
H'FFFF F828	DRLDE00		DRLDE01	
H'FFFF F82C	DRLDE02		DRLDE03	
H'FFFF F830	TCNTE00		TCNTE01	
H'FFFF F834	TCNTE02		TCNTE03	
H'FFFF F838	PSCCRE00	PSCCRE01	PSCCRE02	PSCCRE03
H'FFFF F83C	Reserved			
H'FFFF F840	TCRE1	TOCRE1	TIERE1	RLDCRE1
H'FFFF F844	TSRE1	Reserved		
H'FFFF F848	PSCRE1	Reserved		
H'FFFF F84C	SSTRE1	Reserved		
H'FFFF F850	CYLRE10		CYLRE11	
H'FFFF F854	CYLRE12		CYLRE13	
H'FFFF F858	DTRE10		DTRE11	
H'FFFF F85C	DTRE12		DTRE13	
H'FFFF F860	CRLDE10		CRLDE11	
H'FFFF F864	CRLDE12		CRLDE13	

Address	31 24	23 16	15 8	7 0
H'FFFF F868	DRLDE10		DRLDE11	
H'FFFF F86C	DRLDE12		DRLDE13	
H'FFFF F870	TCNTE10		TCNTE11	
H'FFFF F874	TCNTE12		TCNTE13	
H'FFFF F878	PSCCRE10	PSCCRE11	PSCCRE12	PSCCRE13
H'FFFF F87C	Reserved			
H'FFFF F880	TCRE2	TOCRE2	TIERE2	RLDCRE2
H'FFFF F884	TSRE2	Reserved		
H'FFFF F888	PSCRE2	Reserved		
H'FFFF F88C	SSTRE2	Reserved		
H'FFFF F890	CYLRE20		CYLRE21	
H'FFFF F894	CYLRE22		CYLRE23	
H'FFFF F898	DTRE20		DTRE21	
H'FFFF F89C	DTRE22		DTRE23	
H'FFFF F8A0	CRLDE20		CRLDE21	
H'FFFF F8A4	CRLDE22		CRLDE23	
H'FFFF F8A8	DRLDE20		DRLDE21	
H'FFFF F8AC	DRLDE22		DRLDE23	
H'FFFF F8B0	TCNTE20		TCNTE21	
H'FFFF F8B4	TCNTE22		TCNTE23	
H'FFFF F8B8	PSCCRE20	PSCCRE21	PSCCRE22	PSCCRE23
H'FFFF F8BC	Reserved			
H'FFFF F8C0	TCRE3	TOCRE3	TIERE3	RLDCRE3
H'FFFF F8C4	TSRE3	Reserved		
H'FFFF F8C8	PSCRE3	Reserved		
H'FFFF F8CC	SSTRE3	Reserved		
H'FFFF F8D0	CYLRE30		CYLRE31	
H'FFFF F8D4	CYLRE32		CYLRE33	
H'FFFF F8D8	DTRE30		DTRE31	
H'FFFF F8DC	DTRE32		DTRE33	
H'FFFF F8E0	CRLDE30		CRLDE31	
H'FFFF F8E4	CRLDE32		CRLDE33	

Address	31 24	23 16	15 8	7 0
H'FFFF F8E8	DRLDE30		DRLDE31	
H'FFFF F8EC	DRLDE32		DRLDE33	
H'FFFF F8F0	TCNTE30		TCNTE31	
H'FFFF F8F4	TCNTE32		TCNTE33	
H'FFFF F8F8	PSCCRE30	PSCCRE31	PSCCRE32	PSCCRE33
H'FFFF F8FC	Reserved			
H'FFFF F900	TCRE4	TOCRE4	TIERE4	RLDCRE4
H'FFFF F904	TSRE4	Reserved		
H'FFFF F908	PSCRE4	Reserved		
H'FFFF F90C	SSTRE4	Reserved		
H'FFFF F910	CYLRE40		CYLRE41	
H'FFFF F914	CYLRE42		CYLRE43	
H'FFFF F918	DTRE40		DTRE41	
H'FFFF F91C	DTRE42		DTRE43	
H'FFFF F920	CRLDE40		CRLDE41	
H'FFFF F924	CRLDE42		CRLDE43	
H'FFFF F928	DRLDE40		DRLDE41	
H'FFFF F92C	DRLDE42		DRLDE43	
H'FFFF F930	TCNTE40		TCNTE41	
H'FFFF F934	TCNTE42		TCNTE43	
H'FFFF F938	PSCCRE40	PSCCRE41	PSCCRE42	PSCCRE43
H'FFFF F93C	Reserved			

Address	**31 24**	**23 16**	**15 8**	**7 0**
H'FFFF F940	TCRE5	TOCRE5	TIERE5	RLDCRE5
H'FFFF F944	TSRE5	Reserved		
H'FFFF F948	PSCRE5	Reserved		
H'FFFF F94C	SSTRE5	Reserved		
H'FFFF F950	CYLRE50		CYLRE51	
H'FFFF F954	CYLRE52		CYLRE53	
H'FFFF F958	DTRE50		DTRE51	
H'FFFF F95C	DTRE52		DTRE53	
H'FFFF F960	CRLDE50		CRLDE51	
H'FFFF F964	CRLDE52		CRLDE53	
H'FFFF F968	DRLDE50		DRLDE51	
H'FFFF F96C	DRLDE52		DRLDE53	
H'FFFF F970	TCNTE50		TCNTE51	
H'FFFF F974	TCNTE52		TCNTE53	
H'FFFF F978	PSCCRE50	PSCCRE51	PSCCRE52	PSCCRE53
H'FFFF F97C	Reserved			
H'FFFF F980	TCRE6	TOCRE6	TIERE6	RLDCRE6
H'FFFF F984	TSRE6	Reserved		
H'FFFF F988	PSCRE6	Reserved		
H'FFFF F98C	SSTRE6	Reserved		
H'FFFF F990	CYLRE60		CYLRE61	
H'FFFF F994	CYLRE62		CYLRE63	
H'FFFF F998	DTRE60		DTRE61	
H'FFFF F99C	DTRE62		DTRE63	
H'FFFF F9A0	CRLDE60		CRLDE61	

Address	31 24	23 16	15 8	7 0
H'FFFF F9A4	CRLDE62		CRLDE63	
H'FFFF F9A8	DRLDE60		DRLDE61	
H'FFFF F9AC	DRLDE62		DRLDE63	
H'FFFF F9B0	TCNTE60		TCNTE61	
H'FFFF F9B4	TCNTE62		TCNTE63	
H'FFFF F9B8	PSCCRE60	PSCCRE61	PSCCRE62	PSCCRE63
H'FFFF F9BC	Reserved			
H'FFFF F9FC	Reserved			

Table 13.9 Timer F Registers

Address	31 24	23 16	15 8	7 0
H'FFFF FA00	TSTRF			
H'FFFF FA04	NCCRF			
H'FFFF FA08	Reserved			
H'FFFF FA0C	NCMCR1F			
H'FFFF FA10	NCNTFA0	NCRFA0	NCNTFA1	NCRFA1
H'FFFF FA14	NCNTFA2	NCRFA2	NCNTFA3	NCRFA3
H'FFFF FA18	NCNTFA4	NCRFA4	NCNTFA5	NCRFA5
H'FFFF FA1C	NCNTFA6	NCRFA6	NCNTFA7	NCRFA7
H'FFFF FA20	NCNTFA8	NCRFA8	NCNTFA9	NCRFA9
H'FFFF FA24	NCNTFA10	NCRFA10	NCNTFA11	NCRFA11
H'FFFF FA28	NCNTFA12	NCRFA12	NCNTFA13	NCRFA13
H'FFFF FA2C	NCNTFA14	NCRFA14	NCNTFA15	NCRFA15
H'FFFF FA30	NCNTFA16	NCRFA16	NCNTFA17	NCRFA17
H'FFFF FA34	NCNTFA18	NCRFA18	NCNTFA19	NCRFA19
H'FFFF FA38	NCNTFA20	NCRFA20	NCNTFA21	NCRFA21
H'FFFF FA3C	NCNTFA22	NCRFA22	NCNTFA23	NCRFA23
H'FFFF FA40	NCNTFA24	NCRFA24	NCNTFA25	NCRFA25
H'FFFF FA44	NCNTFA26	NCRFA26	NCNTFA27	NCRFA27
H'FFFF FA4C	Reserved			
H'FFFF FA50	NCNTFB0	NCRFB0	NCNTFB1	NCRFB1
H'FFFF FA54	NCNTFB2	NCRFB2	Reserved	
H'FFFF FA58 : H'FFFF FA7C	Reserved			
H'FFFF FA80	TCRF0	TIERF0	Reserved	TSRF0
H'FFFF FA84	ECNTAF0			
H'FFFF FA88	ECNTBF0		GRBF0	
H'FFFF FA8C	ECNTCF0			
H'FFFF FA90	GRAF0			
H'FFFF FA94	CDRF0			
H'FFFF FA98	GRCF0			

Address	31 24	23 16	15 8	7 0
H'FFFF FA9C	Reserved			
H'FFFF FAA0 : H'FFFF FBFC	(F01 to F11)			
H'FFFF FC00	TCRF12	TIERF12	Reserved	TSRF12
H'FFFF FC04	ECNTAF12			
H'FFFF FC08	ECNTBF12		GRBF12	
H'FFFF FC0C	ECNTCF12			
H'FFFF FC10	GRAF12			
H'FFFF FC14	CDRF12			
H'FFFF FC18	GRCF12			
H'FFFF FC1C	GRDF12 (only F12 to F15)			
H'FFFF FC20 : H'FFFF FC7C	(F13 to F15)			
H'FFFF FC80	TCRF16	TIERF16	Reserved	TSRF16
H'FFFF FC84	ECNTAF16			
H'FFFF FC88	ECNTBF16		GRBF16	
H'FFFF FC8C	ECNTCF16			
H'FFFF FC90	GRAF16			
H'FFFF FC94	CDRF16			
H'FFFF FC98	GRCF16			
H'FFFF FC9C	Reserved			
H'FFFF FCA0 : H'FFFF FDFC	(F17 to F27)			

Table 13.10 Timer G Registers

Address	31 24	23 16	15 8	7 0
H'FFFF FE00	Reserved	TSTRG	Reserved	
H'FFFF FE04 : H'FFFF FE7C	Reserved			
H'FFFF FE80	TCRG0	TSRG0	Reserved	
H'FFFF FE84	TCNTG0		OCRG0	
H'FFFF FE88	Reserved			
H'FFFF FE8C				
H'FFFF FE90	TCRG1	TSRG1	Reserved	
H'FFFF FE94	TCNTG1		OCRG1	
H'FFFF FE98	Reserved			
H'FFFF FE9C				
H'FFFF FEA0	TCRG2	TSRG2	Reserved	
H'FFFF FEA4	TCNTG2		OCRG2	
H'FFFF FEA8	Reserved			
H'FFFF FEAC				
H'FFFF FEB0	TCRG3	TSRG3	Reserved	
H'FFFF FEB4	TCNTG3		OCRG3	
H'FFFF FEB8	Reserved			
H'FFFF FEBC				
H'FFFF FEC0	TCRG4	TSRG4	Reserved	
H'FFFF FEC4	TCNTG4		OCRG4	
H'FFFF FEC8	Reserved			
H'FFFF FECC				
H'FFFF FED0	TCRG5	TSRG5	Reserved	
H'FFFF FED4	TCNTG5		OCRG5	
H'FFFF FED8 : H'FFFF FEFC	Reserved			

Table 13.11 Timer H Registers

Address	31 24	23 16	15 8	7 0
H'FFFF FF00 : H'FFFF FF3C	Reserved			
H'FFFF FF40	TCRH	TSRH	Reserved	
H'FFFF FF44	TCNT1H		OCR1H	
H'FFFF FF48	TCNT2H			
H'FFFF FF4C : H'FFFF FF7C	Reserved			

Table 13.12 Timer J Registers

Address	31 24	23 16	15 8	7 0
H'FFFF FF80	TSTRJ	Reserved		
H'FFFF FF84 : H'FFFF FF8C	Reserved			
H'FFFF FF90	TCRJ0	FCRJ0	TSRJ0	Reserved
H'FFFF FF94	TIERJ0	FDNRJ0	NCNTJ0	NCRJ0
H'FFFF FF98	TCNTJ0		OCRJ0	
H'FFFF FF9C	FIFOJ0		Reserved	
H'FFFF FFA0	TCRJ1	FCRJ1	TSRJ1	Reserved
H'FFFF FFA4	TIERJ1	FDNRJ1	NCNTJ1	NCRJ1
H'FFFF FFA8	TCNTJ1		OCRJ1	
H'FFFF FFAC	FIFOJ1		Reserved	
H'FFFF FFB0 : H'FFFF FFFC	Reserved			

13.3 Input/Output Pins

Table 13.13 Pin Configuration

Block	Symbol	I/O	Function
Common controller	TCLKA	Input	Input pin for the external clock signal to be supplied on signal line 4 of the clock bus
	TCLKB	Input	Input pin for the external clock signal that can be supplied on signal line 5 of the clock bus
Timer A	TIA00 to TIA05	Input	Input pins for input-capture triggers for timer A channels
Timer C	TIOC00 to TIOC03, TIOC10 to TIOC13, TIOC20 to TIOC23, TIOC30 to TIOC33, TIOC40 to TIOC43	I/O	Input pins for input-capture triggers and output pins for output-compare signals of timer C (one each for channels 0 to 3 in subblocks C0 to C4)
Timer D	TOD00A to TOD03A, TOD10A to TOD13A, TOD20A to TOD23A, TOD30A to TOD33A	Output	Output pins for compare-match signals of timer D (one each for channels 0 to 3 in subblocks D0 to D3)
	TOD00B to TOD03B, TOD10B to TOD13B, TOD20B to TOD23B, TOD30B to TOD33B	Output	Output pins for one-shot pulses of timer D (one each for channels 0 to 3 in subblocks D0 to D3)
Timer E	TOE00 to TOE03, TOE10 to TOE13, TOE20 to TOE23, TOE30 to TOE33, TOE40 to TOE43, TOE50 to TOE53, TOE60 to TOE63	Output	Output pins for PWM signals of timer E (one each for channels 0 to 3 in subblocks E0 to E6)
Timer F	TIF0A to TIF2A, TIF3 to TIF27	Input	Input pins for event signals for subblocks F0 to F19 of timer F TIF0A to TIF2A: Input pins for subblocks F0 to F2 TIF3 to TIF27: Input pins for subblocks F3 to F27
	TIF0B to TIF2B	Input	Input pins for event signals for subblocks F0 to F2 of timer F
Timer J	TIJ0 and TIJ1	Input	Input pins for input-capture triggers for subblocks of timer J

13.4 Overview of Common Controller

The common controller controls the ATU-III module as a whole. For example, it enables and disables the prescalers and timer counters for timers A to J and controls the clock bus.

13.4.1 Clock Bus

The clock bus consists of six signal lines used to distribute the source signals for counting (count enabling signals) to the timer channels. The timer counters on each of the channels run in synchronization with the internal peripheral clock (Pϕ).

Table 13.14 shows the signals which are available for input on the clock bus.

Table 13.14 Signals to be Input on Clock Bus

Bit Number of Clock Bus	Input Signals
5	Output signal from timer B (multiplied-and-corrected clock signal) or external input clock B (TCLKB)
4	External input clock A (TCLKA)
3	Output signal from prescaler 3
2	Output signal from prescaler 2
1	Output signal from prescaler 1
0	Output signal from prescaler 0

13.5 Register Description of Common Controller

13.5.1 ATU-III Master Enable Register (ATUENR)

ATUENR is a 16-bit readable/writable register. This register is used to enable and disable the prescalers and the individual timers in ATU-III. Setting an enable bit to 1 enables the corresponding timer. Clearing the bit to 0 disables the corresponding timer. Even when the enable bit is cleared to 0, the registers of the corresponding timer remain accessible.

Timers can be synchronized by simultaneously setting multiple bits to 1. Note that a particular subblock cannot be synchronized with other subblocks while they are counting.

ATUENR can be read from or written to in byte or word units.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	TJE	THE	TGE	TFE	TEE	TDE	TCE	TBE	TAE	PSCE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	TJE	0	R/W	Timer J Enable Enables and disables counter operation of timer J. When the counter is disabled, its value is retained. When this bit is again set to 1, the counter resumes counting from the retained value. However, the corresponding bit in the timer J start register must also be set to 1 to enable the operation of either of the subblock counters. 0: Timer J counter operation disabled 1: Timer J counter operation enabled

Bit	Bit Name	Initial Value	R/W	Description
8	THE	0	R/W	Timer H Enable Enables and disables counter operation of timer H. When the counter is disabled, its value is retained. When this bit is again set to 1, the counter resumes counting from the retained value. 0: Timer H counter operation disabled 1: Timer H counter operation enabled
7	TGE	0	R/W	Timer G Enable Enables and disables counter operation of timer G. When the counter is disabled, its value is retained. When this bit is again set to 1, the counter resumes counting from the retained value. However, the corresponding bit in the timer G start register must also be set to 1 to enable the operation of either of the subblock counters. 0: Timer G counter operation disabled 1: Timer G counter operation enabled
6	TFE	0	R/W	Timer F Enable Enables and disables counter operation of timer F. When the counter is disabled, its value is retained. When this bit is again set to 1, the counter resumes counting from the retained value. However, the corresponding bit in the timer F start register must also be set to 1 to enable the operation of either of the subblock counters. 0: Timer F counter operation disabled 1: Timer F counter operation enabled
5	TEE	0	R/W	Timer E Enable Enables and disables counter operation of timer E. When the counter is disabled, its value is retained. When this bit is again set to 1, the counter resumes counting from the retained value. However, the corresponding bit in the timer E start register must also be set to 1 to enable the operation of either of the subblock counters. 0: Timer E counter operation disabled 1: Timer E counter operation enabled

Bit	Bit Name	Initial Value	R/W	Description
4	TDE	0	R/W	Timer D Enable Enables and disables counter operation of timer D. When the counter is disabled, its value is retained. When this bit is again set to 1, the counter resumes counting from the retained value. However, the corresponding bit in the timer D start register must also be set to 1 to enable the operation of either of the subblock counters. 0: Timer D counter operation disabled 1: Timer D counter operation enabled
3	TCE	0	R/W	Timer C Enable Enables and disables counter operation of timer C. When the counter is disabled, its value is retained. When this bit is again set to 1, the counter resumes counting from the retained value. However, the corresponding bit in the timer C start register must also be set to 1 to enable the operation of either of the subblock counters. 0: Timer C counter operation disabled 1: Timer C counter operation enabled
2	TBE	0	R/W	Timer B Enable Enables and disables counter operation of timer B. When the counter is disabled, its value is retained. When this bit is again set to 1, the counter resumes counting from the retained value. 0: Timer B counter operation disabled 1: Timer B counter operation enabled
1	TAE	0	R/W	Timer A Enable Enables and disables counter operation of timer A. When the counter is disabled, its value is retained. When this bit is again set to 1, the counter resumes counting from the retained value. 0: Timer A counter operation disabled 1: Timer A counter operation enabled

Bit	Bit Name	Initial Value	R/W	Description
0	PSCE	0	R/W	Prescaler Enable Enables and disables the prescaler counters. When the prescaler counters are disabled, the counter values are retained. Once the bit is set to 1 again, the counter resumes counting from the retained value. 0: Prescaler counter operation disabled 1: Prescaler counter operation enabled

13.5.2 Clock Bus Control Register (CBCNT)

CBCNT is an 8-bit readable/writable register that selects the source of the clock signal to be supplied on signal line 5 of the clock bus and the valid edge of external clock signals (only applies to line 5 when the external input clock is selected).

CBCNT can be read from or written to in bytes.

Bit:	7	6	5	4	3	2	1	0
	-	-	CB4EG[1:0]		-	CB5 SEL	CB5EG[1:0]	
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5, 4	CB4EG [1:0]	00	R/W	Clock Bus 4 Edge Select These bits select the edge sense for external input clock A (TCLKA). The clock signal is output on signal line 4 of the clock bus. Counters for which signal line 4 of the clock bus has been selected as the source for counting count on the edge selected by these bits. 00: Neither edge of the external clock is sensed 01: Rising edges of the external clock are sensed 10: Falling edges of the external clock are sensed 11: Both rising and falling edges of the external clock are sensed

Bit	Bit Name	Initial Value	R/W	Description
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	CB5SEL	0	R/W	Clock Bus 5 Source Select Selects the source of the clock to be output on signal line 5 of the clock bus. 0: External input clock B (TCLKB) 1: Multiplied-and-corrected clock output by Timer B
1, 0	CB5EG [1:0]	00	R/W	Clock Bus 5 Edge Select These bits select the edge sense for external input clock B (TCLKB). The clock signal is output on signal line 5 of the clock bus. Counters for which signal line 5 of the clock bus has been selected as the source for counting count on the edge selected by these bits. The setting of these bits is only valid when the TCLKB signal is selected as the source for line 5 of the clock bus. When the multiplied-and-corrected clock is selected as the source for line 5 of the clock bus, the setting of these bits is invalid. 00: Neither edge of the external clock is sensed 01: Rising edges of the external clock are sensed 10: Falling edges of the external clock are sensed 11: Both rising and falling edges of the external clock are sensed

13.5.3 Noise Cancellation Mode Register (NCMR)

NCMR is an 8-bit readable/writable register that selects the mode and clock to drive the counter for of the noise canceler in each of timers A, C, F, and J.

In premature-transition cancellation mode, subsequent changes to the input signal level are ignored if they come within a given period of a detected change. That is, level changes within a certain period of an initial one are treated as noise.

In minimum time-at-level cancellation mode, the first and subsequent level changes are ignored unless the input signal level remains the same over a given period. Level changes occurring within a shorter period are considered to indicate an unstable signal, and such signals are treated as noise.

The period is set by noise canceler registers in each of the applicable blocks (i.e. in timers A, C, F, and J) and is counted by a noise canceler counter.

Figures 13.1 and 13.2 show the operation of the premature-transition cancellation and minimum time-at-level cancellation, using the TIA00 input signal of timer A as an example.

The edge for counting is detected from signals after noise removal in timers A, C, F, and J. Rising edges are being detected in figures 13.1 and 13.2.

NCMR can be read from and written to in byte units.

Bit:	7	6	5	4	3	2	1	0
	NCC SEL	-	-	-	NCMJ	NCMF	NCMC	NCMA
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	NCCSEL	0	R/W	Noise Canceler Counter Clock Select Selects the clock for counting by the noise cancelers. The peripheral clock (Pϕ) or Pϕ divided by 128 can be selected. The default setting is the clock divided by 128. The same counter clock must be used for all timers other than timer A. In the case of timer A, the clock signal on clock-bus line 5 is also available. For details, see section 13.10.4, Timer I/O Control Register 2A (TIOR2A). 0: Pϕ divided by 128 is used as the counter clock 1: Pϕ is used as the counter clock
6 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	NCMJ	0	R/W	Timer J Noise Cancellation Mode Selects the noise cancellation mode for timer J. The same mode is used on both channels of timer J. 0: Premature-transition cancellation mode 1: Minimum time-at-level cancellation mode
2	NCMF	0	R/W	Timer F Noise Cancellation Mode Selects the noise cancellation mode for timer F. The same mode is used on both channels of timer F. Modes of operation are individually selectable for each channel of Timer F. To make per-channel settings, start by setting this bit to 0 as the setting for Timer F as a unit, and then set noise cancellation mode channel register 1F (NCMCR1F). 0: Premature-transition cancellation mode 1: Minimum time-at-level cancellation mode
1	NCMC	0	R/W	Timer C Noise Cancellation Mode Selects the noise cancellation mode for timer C. The same mode is used on both channels of timer C. 0: Premature-transition cancellation mode 1: Minimum time-at-level cancellation mode

Bit	Bit Name	Initial Value	R/W	Description
0	NCMA	0	R/W	Timer A Noise Cancellation Mode Selects the noise cancellation mode for timer A. The same mode is used on both channels of timer A. Modes of operation are individually selectable for each channel of Timer A. To make per-channel settings, start by setting this bit to 0 as the setting for Timer A as a unit, and then set noise cancellation mode channel register 1A (NCMCR1A). 0: Premature-transition cancellation mode 1: Minimum time-at-level cancellation mode

Table 13.15 shows the truth table for settings that determine the noise cancellation mode.

Table 13.15 Settings that Determine Noise Cancellation Modes for Timer A

Channel enable	Filter mode	Channel register	Mode of cancellation	Unit of filtering
TIOR2A.NCEAn	NCMR.NCMA	NCMCR1A.NCM1An		
0	—	—	Filter invalid	—
1	0	0	Minimum time at level	Individual subblock/channel
1	0	1	Premature-transition	Individual channel
1	1	—	Premature-transition	Individual subblock

Note: n = 0 to 5
Enable bit (NCEAn) can be made per channel.

Methods of setting noise cancellation for timer A according to the unit of filtering (precondition: channel enable = 1)

- Settings for individual channels: After setting the noise cancellation mode bit (NCMA) in the noise cancellation mode register (NCMR) to 0, set even one of the noise cancellation mode channel setting bits (NCM1An) in noise cancellation mode channel register 1A (NCMCR1A) of timer A to 1.

- Setting for all channels: Set the noise cancellation mode bit (NCMA, the same bit as in "Settings for individual channels" above) in the noise cancellation mode register (NCMR) to 1 or set all of the noise cancellation mode channel setting bits (NCM1An) in noise cancellation mode channel register 1A (NCMCR1A) to 0.

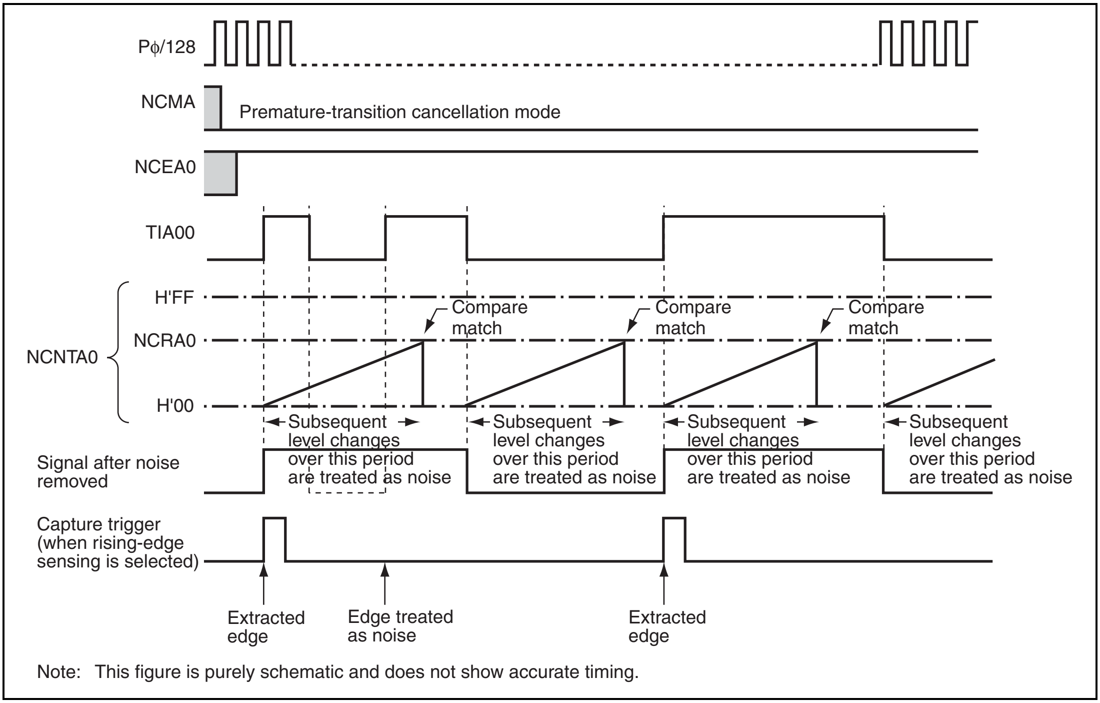
Table 13.16 Settings that Determine Noise Cancellation Modes for Timer F

Channel enable	Filter mode	Channel register	Mode of	
NCCRF.NCEFn	NCMR.NCMF	NCMCR1F.NCM1Fn	cancellation	Unit of filtering
0	—	—	Filter invalid	—
1	0	0	Premature-transition	Each subblock/channel
1	0	1	Minimum time at level	Each channel
1	1	—	Minimum time at level	Each subblock

Note: n = 0 to 27
Enable bit (NCEFn) settings can be made per channel.

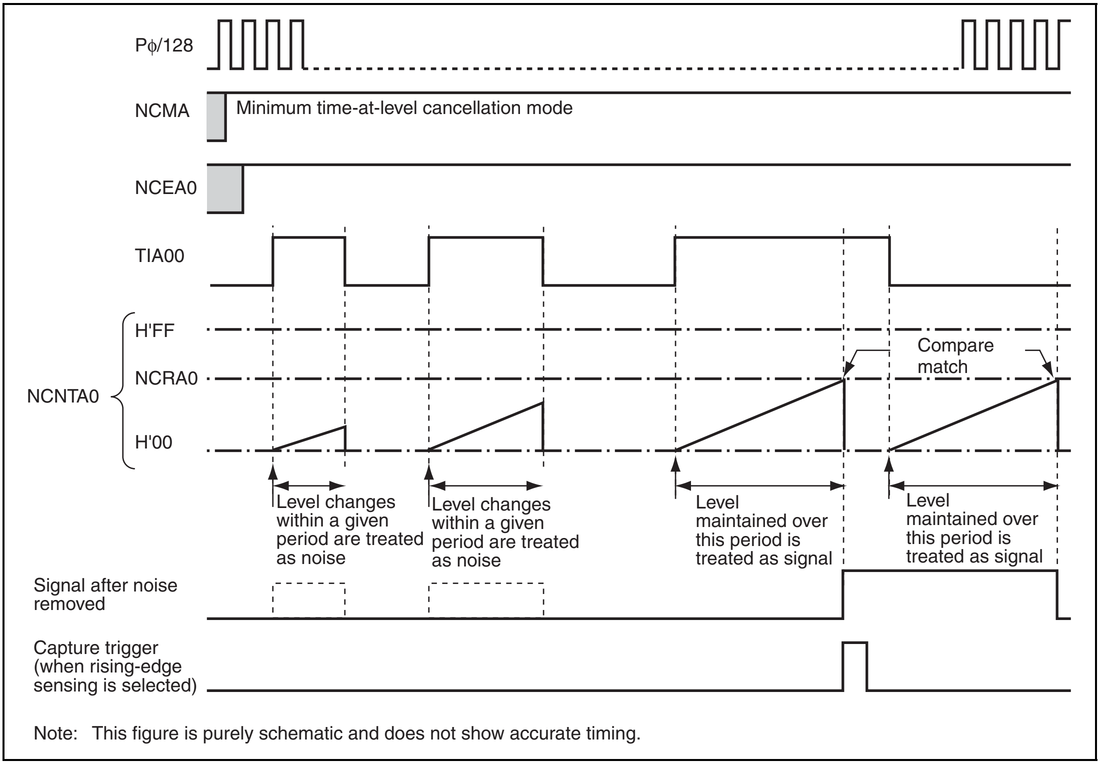
Methods of setting noise cancellation for timer F according to the unit of filtering (precondition: channel enable = 1)

- Settings for individual channels: After setting the noise cancellation mode bit (NCMF) in the noise cancellation mode register (NCMR) to 0, set even one of the noise cancellation mode channel setting bits (NCM1Fn) in noise cancellation mode channel register 1F (NCMCR1F) of timer A to 1.
- Setting for all channels: Set the noise cancellation mode bit (NCMF, the same bit as in ”Settings for individual channels” above) in the noise cancellation mode register (NCMR) to 1 or set all of the noise cancellation mode channel setting bits (NCM1Fn) in noise cancellation mode channel register 1F (NCMCR1F) to 0.


Pϕ/128
NCMA
Premature-transition cancellation mode
NCEA0
TIA00
H'FF
NCRA0
NCNTA0
H'00
Compare match
Compare match
Compare match
Subsequent level changes over this period are treated as noise
Subsequent level changes over this period are treated as noise
Subsequent level changes over this period are treated as noise
Subsequent level changes over this period are treated as noise
Signal after noise removed
Capture trigger (when rising-edge sensing is selected)
Extracted edge
Edge treated as noise
Extracted edge

Note: This figure is purely schematic and does not show accurate timing.

Figure 13.1 Operation of Noise Canceler in Premature-Transition Cancellation Mode


Pϕ/128
NCMA
Minimum time-at-level cancellation mode
NCEA0
TIA00
NCNTA0
H'FF
NCRA0
H'00
Compare match
Level changes within a given period are treated as noise
Level changes within a given period are treated as noise
Level maintained over this period is treated as signal
Level maintained over this period is treated as signal
Signal after noise removed
Capture trigger (when rising-edge sensing is selected)

Note: This figure is purely schematic and does not show accurate timing.

Figure 13.2 Operation of Noise Canceler in Minimum-Time-at Level Cancellation Mode

13.6 Overview of Prescalers

ATU-III includes four general prescalers and a prescaler for the noise-canceler clock.

The general prescalers are implemented as 10-bit down-counters, in which the prescaled clock signals are derived by frequency-dividing the peripheral clock (Pϕ) by N ($1 \leq N \leq 1024$).

The division ratio is obtained from the following expression.

$$\text{Division ratio of prescaler} = \frac{1}{\text{PSCn[9:0]} + 1} \text{ (Settable value: 1/1 to 1/1024)}$$

A duty cycle of 50% is not guaranteed for the clock-signal outputs of the prescalers. Instead, the high level is output in one cycle of the Pϕ clock and the low level is output in the remaining cycles within the prescaled period. When 1/1 is selected as the division ratio, the high level is always output on the clock bus.

The generated clock signals are supplied to the individual timers via the clock bus. The prescalers for each of the channels operate independently. The prescalers can, however, be started in synchronization with each other after a reset by setting the PSCE bit in ATUENR to 1 after the appropriate values have been set. Synchronization of a prescaler is not possible after it has started or its division ratio has been changed.

The prescaler for the noise-canceler clock is implemented as a 7-bit down-counter. This generates a clock signal by frequency-dividing the peripheral clock (Pϕ) by 128. The clock signal thus generated is supplied to the noise cancelers of timers A, C, F, and J.

The peripheral clock of the peripheral clock frequency divided by 128 can be selected for the noise-canceler clock by the NCCSEL bit in NCMR of the common controller. Further division ratios are not available.

The down-counters of the prescalers are initialized to H'000 by a power-on reset or a transition to the hardware standby mode.

13.7 Register Description of Prescalers

13.7.1 Prescaler Registers 0 to 3 (PSCR0 to PSCR3)

PSCR0 to PSCR3 are 16-bit readable/writable registers, each of which holds a division ratio for one of the four prescalers.

After a prescaler counter underflows, counting restarts from the setting in this register. Settable values range from H'000 to H'3FF.

PSCR0 to PSCR3 can be read from or written to in word units.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	PSCn[9:0]									
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: n = 0 to 3

Bit	Bit Name	Initial Value	R/W	Description
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 0	PSCn[9:0]	All 0	R/W	Division Ratio These bits specify the division ratio for the corresponding prescaler.

13.8 Operation of Prescalers

13.8.1 Starting Prescalers

The prescalers start operating when the PSCE bit in the ATU-III master enable register (ATUENR) is set to 1 and generates a clock with a frequency given by the division ratio in the PSCn bits. While a prescaler is operating, the high level is output for one cycle of the Pϕ clock each time the corresponding prescaler counter underflows.

When the setting in the PSCn bits is changed during operation, the division ratio of the output clock is updated on the first subsequent counter underflow.

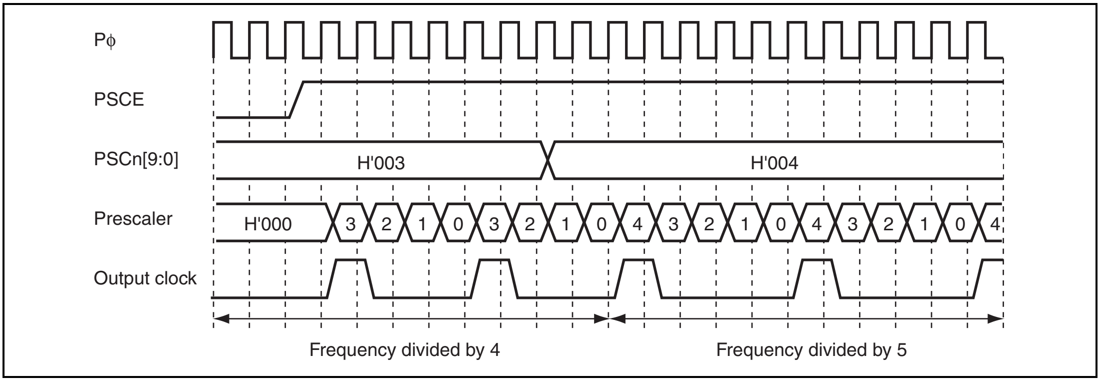


Figure 13.3 Starting Prescaler

13.8.2 Stopping and Restarting Operation

The prescaler stops operating when the PSCE bit in the ATU-III master enable register (ATUENR) is cleared to 0. The clock signal stays at the low level and the value in the prescaler counter is retained while the prescaler is stopped. Setting the PSCE bit to 1 makes counting restart from the retained value.

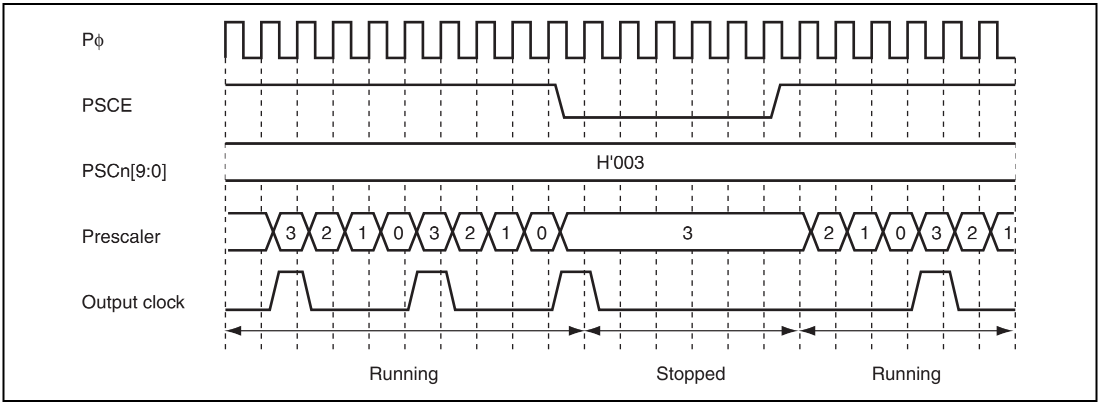


Figure 13.4 Stopping Prescaler

13.9 Overview of Timer A

Timer A includes a free-running counter A (TCNTA) and input capture registers A0 to A5 (ICRA0 to ICRA5). TCNTA is a free-running up counter. An interrupt request can be output when the counter overflows.

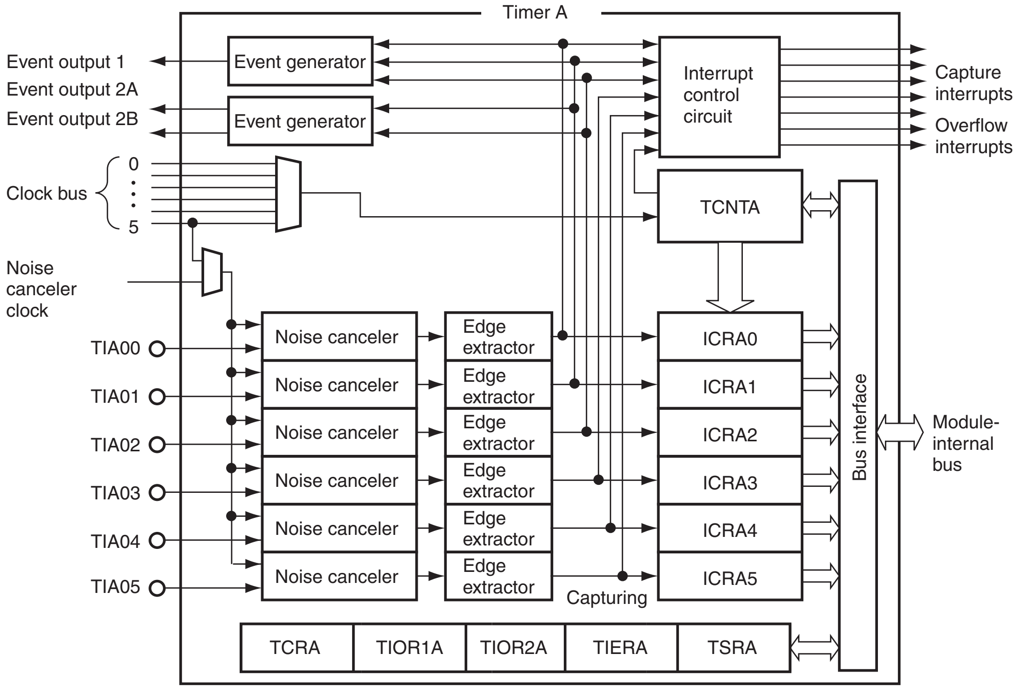
Input capture registers A0 to A5 (ICRA0 to ICRA5) capture the value of free-running counter A (TCNTA) on an assertion of the corresponding external input signal (TIA00 to TIA05). The rising or falling edge, or both edges, can be selected as the trigger for capture. The edge selection is made timer I/O control register 1A. The interrupt is requested or the A-DMAC is activated at the same timing as capturing.

Noise on the external input signals can be removed by the noise canceler. Signals on pins TIA00 to TIA02 can be output to timer B or D as event signals after noise has been removed and their edges have been extracted. One of TIA00 to TIA02 are selectable for output to timer B as an event signal (event output 1). TIA01 and TIA02 are selectable for output to timers D0 to D3 (for timers D0 to D2 as event output 2A and for timer D3 as event output 2B) and these event outputs can be used as capture triggers for timer offset base registers for D0 to D3 (OSBRD0 to OSBRD3).

When the A-DMAC is activated by a capture operation, it reads the capture register associated with the interrupt source and transfers the value read from the register and any previously captured value stored in the A-DMAC buffer register to the on-chip RAM.

13.9.1 Block Diagram of Timer A

Figure 13.5 is a block diagram of timer A.



[Legend]

TCRA:	Timer control register A
TIOR1A:	Timer I/O control register 1A
TIOR2A:	Timer I/O control register 2A
TIERA:	Timer interrupt enable register A
TSRA:	Timer status register A
ICRA:	Input capture register A
TCNTA:	Free-running counter A

Figure 13.5 Block Diagram of Timer A

13.10 Description of Timer A Registers

13.10.1 Timer Control Register A (TCRA)

TCRA is an 8-bit readable/writable register that sets the event output generated from external input signals (TIA00 to TIA02) and the counter clock.

TCRA can be read from and written to in byte or word units.

TCRA is initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	EVO SEL2A	EVO SEL2B	EVOSEL1			CKSELA		
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	EVOSEL2A	0	R/W	Event Output 2A Select Selects the signal for output on event output 2A as the externally input signal TIA01 or TIA02 (after the removal of noise and edge extraction). In timers D0 to D2, the event output 2A signal can be used as a capture trigger for the corresponding timer-offset base registers (OSBRD0 to OSBRD2). 0: TIA01 is selected as event output 2A 1: TIA02 is selected as event output 2A
6	EVOSEL2B	0	R/W	Event Output 2B Select Selects the signal for output on event output 2B as the externally input signal TIA01 or TIA02 (after the removal of noise and edge extraction). In timer D3, the event output 2B signal can be used as a capture trigger for the corresponding timer-offset base register (OSBRD3). 0: TIA01 is selected as event output 2B 1: TIA02 is selected as event output 2B

Bit	Bit Name	Initial Value	R/W	Description
5 to 3	EVOSEL1	000	R/W	Event Output 1 Select These bits select the signal for output on event output 1 as one of the externally input signal TIA00 to TIA02 (after the removal of noise and edge extraction). In timer B, the event output 1 signal can be used as an event output 1 to timer B. 000: No signal is selected as event output 1 001: TIA00 is selected as event output 1 010: TIA01 is selected as event output 1 011: Setting prohibited 100: TIA02 is selected as event output 1 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
2 to 0	CKSELA	000	R/W	Clock Select A These bits select the signal on one of clock-bus lines 0 to 5 as the clock signal for counting. The signal on lines 0 to 3 have been frequency-divided by prescalers 0 to 3, respectively. Clock-bus line 4 supplies externally input clock A (TCLKA). Clock-bus line 5 supplies externally input clock B (TCLKB) or the multiplied-and-corrected clock output by timer B. Stop timer A before making or changing the counter-clock selection. 000: Clock-bus line 0 (prescaler 0) 001: Clock-bus line 1 (prescaler 1) 010: Clock-bus line 2 (prescaler 2) 011: Clock-bus line 3 (prescaler 3) 100: Clock-bus line 4 (TCLKA) 101: Clock-bus line 5 (TCLKB or multiplied-and-corrected clock) 110: Setting prohibited 111: Setting prohibited

Note: The edge of an external input clock is extracted before it is output on a clock bus. When using external input clock A or B, select the edge to be extracted by setting the CB4EG and CB5EG bits in the clock bus control register (CBCNT).

13.10.2 Noise cancellation mode Channel Register 1A (NCMCR1A)

NCMCR1A is an 8-bit readable/writable register that selects the operation mode of noise canceller for each channel unit.
In the premature-transition cancellation mode, after a change in the level of the input signal has been detected, further changes in the level of the input signal that occur within a specified period are ignored. That is, changes with a specified period after each initial level change are regarded as noise in this mode.

The periods are set in the noise canceler registers in each of the applicable blocks, and are counted by a noise canceler counters.

Figures 13.1 and 13.2 show examples of the noise cancellation in premature-transition cancellation and minimum time-at-level cancellation modes, respectively. (in these examples, edges are input on pin TIA00 and the rising edge sensing has been selected).

In each channel, only those edges of signals which pass through the noise canceler are detected. Figures 13.1 and 13.2 show examples where rising-edge sensing has been selected for the signal which goes through the noise canceler.

NCMCR1A can be read from and written to in byte or word units.

This register is only effective when the NCMA bit of the noise cancellation mode register (NCMR) in the common control unit is 0. Table 13.5 shows the truth table for setting the noise cancellation mode.

Bit:	7	6	5	4	3	2	1	0
	-	-	NCM 1A5	NCM 1A4	NCM 1A3	NCM 1A2	NCM 1A1	NCM 1A0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	NCM1A5	0	R/W	Noise cancellation mode Bit of Channel n Specify the operation mode of noise canceler in channel n. 0: Premature-transition cancellation mode 1: Minimum time-at-level cancellation mode (When NCMA = 0).
4	NCM1A4	0	R/W	
3	NCM1A3	0	R/W	
2	NCM1A2	0	R/W	
1	NCM1A1	0	R/W	
0	NCM1A0	0	R/W	

Note: n = 0 to 5

13.10.3 Timer I/O Control Register 1A (TIOR1A)

TIOR1A is a 16-bit readable/writable register that sets the edge of external inputs (TIA00 to TIA05) to be extracted.

TIOR1A can be read from and written to in byte or word unit.

TIOR1A is initialized to H'0000 by a power-on reset or a transition to the hardware standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	IOA5		IOA4		IOA3		IOA2		IOA1		IOA0	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11, 10	IOA5	00	R/W	I/O Control An These bits select the edge of external inputs (TIA00 to TIA05) that is to be extracted for use in input-capture triggering. When these bits are set to B'00, input capturing is not performed. When a value other than B'00 is set, the contents of free-running counter A (TCNTA) are transferred to input capture register A (ICRA) on extraction of the selected edge from one of the external inputs. Edge extraction is synchronized with the Pϕ clock. Make sure that the frequency of the Pϕ clock is at least twice the frequency of the external input signal. Otherwise, edge extraction will not performed correctly. When the noise canceler is disabled, the selected edge is simply extracted from the external inputs (TIA00 to TIA05). When the noise canceler is enabled, the selected edge is extracted from the signals after noise removal. Usage of the extracted edges in other timers is selected by the settings of bits EVOSEL1, EVOSEL2A, and EVOSEL2B in TCRA (the signal outputs for other timers as a result of edge extraction are active high). 00: Input capturing is not performed 01: TCNTA is captured in ICRA on the rising edge of TIA 10: TCNTA is captured in ICRA on the falling edge of TIA 11: TCNTA is captured in ICRA on both edges of TIA
9, 8	IOA4	00	R/W	
7, 6	IOA3	00	R/W	
5, 4	IOA2	00	R/W	
3, 2	IOA1	00	R/W	
1, 0	IOA0	00	R/W	

Note: n = 0 to 5

13.10.4 Timer I/O Control Register 2A (TIOR2A)

TIOR2A is a 16-bit readable/writable register that selects the noise canceler clock and enables and disables the noise cancelers for externally input signals (TIA00 to TIA05).

TIOR2A can be read from and written to in byte or word unit.

TIOR2A is initialized to H'0000 by a power-on reset or a transition to the hardware standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	NCKA5	NCKA4	NCKA3	NCKA2	NCKA1	NCKA0	-	-	NCEA5	NCEA4	NCEA3	NCEA2	NCEA1	NCEA0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	NCKA5	0	R/W	Noise Canceler Clock Select An These bits select the signal that drives counting. The noise canceler count clock or the signal on clock-bus line 5 can be selected as the signal for counting. Either the Pϕ clock frequency divided by 128 or the Pϕ clock can be selected as the noise canceler count clock by setting the NCCSEL bit of the common controller. 0: Noise canceler count clock is selected as the signal for counting by NCNTAn 1: Clock-bus line 5 is selected as the signal for counting by NCNTAn
12	NCKA4	0	R/W	
11	NCKA3	0	R/W	
10	NCKA2	0	R/W	
9	NCKA1	0	R/W	
8	NCKA0	0	R/W	
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
5	NCEA5	0	R/W	Noise Canceler Enable An
4	NCEA4	0	R/W	These bits enable and disable the noise cancelers for externally input signals (TIA00 to TIA05) When a level change on externally input signals TIA00 to TIA05 is detected while this bit is set to 1, it is processed in premature-transition cancellation or minimum time-at-level cancellation mode depending on the setting in the noise cancellation mode register (NCMR) of the common controller, or noise cancellation mode channel register 1A (NCMCR1A) of timer A. 0: Noise cancelers for TIA are disabled 1: Noise cancelers for TIA are enabled • In premature-transition cancellation mode When a level change of the externally input signal is detected, the change is output as the signal whose noise is removed and the corresponding noise canceler counter (NCNTA0 to NCNTA5) is started for counting up. Subsequent level changes are masked until the value in the counter reaches the value in the noise canceler register (NCRA0 to NCRA5). The level of the externally input signal is output on this compare match. When these bits are cleared to 0 while the counter (NCNTA0 to NCNTA5) is being incremented, counting continues until the values in the counter and the noise canceler register match. The subsequent level changes are also masked over this period. • In minimum time-at-level cancellation mode When a level change of the externally input signal is detected, the corresponding noise canceler counter (NCNTA0 to NCNTA5) is started for counting up. If subsequent level changes are not detected until the value in the counter reaches the value in the noise canceler register (NCRA0 to NCRA5), the previously accepted level change is output as the signal whose noise is removed on compare match of the counter and noise canceler register. When the subsequent level change is detected before the values in the counter and noise canceler register match, all the changes are treated as noise. Therefore the signal whose noise is removed is not changed. When these bits are cleared to 0 while the counter (NCNTA0 to NCNTA5) is being incremented, counting continues until the values in the counter and the noise canceler register match or a level change on the externally input signal is detected. For details on operations in each mode, see figures 13.1 and 13.2.
3	NCEA3	0	R/W	
2	NCEA2	0	R/W	
1	NCEA1	0	R/W	
0	NCEA0	0	R/W	

Note: n = 0 to 5

13.10.5 Timer Status Register A (TSRA)

TSRA is an 8-bit readable/writable register that indicates overflow on free-running counter A (TCNTA) and input capture on input capture registers A0 to A5 (ICRA0 to ICRA5).

The flags in this register are used as interrupts. When the corresponding bit in timer interrupt enable register A (TIERA) is enabled, setting of a flag can lead to a request DMA transfer by A-DMAC or sending of an interrupt request to the CPU via the A-DMAC.

TSRA can be read from and written to in byte or word units.

TSRA is initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	OVFA	-	ICFA5	ICFA4	ICFA3	ICFA2	ICFA1	ICFA0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only 0 can be written to this bit after it is read as 1 to clear it. Writing 1 is ignored.

Bit	Bit Name	Initial Value	R/W	Description
7	OVFA	0	R/(W)*	Overflow Flag A Indicates that free-running counter A (TCNTA) has overflowed. This flag cannot be set to 1 by software. To clear this bit, write 0 to it after reading it as 1. Writing 0 before reading it as 1 has no effect. 0: Indicates that TCNTA has not overflowed [Clearing condition] • When writing 0 to this bit after reading it as 1 1: Indicates that TCNTA has overflowed [Setting condition] • When TCNTA overflows (transition from H'FFFFFFFF to H'00000000)
6	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
5	ICFA5	0	R/(W)*	Input Capture Flag An These bits indicate that the value in free-running counter A (TCNTA) has been captured by input capture register An (ICRAn). When one of these bits is read as 1, the value in TCNTA has been stored in the corresponding ICRAn. These bits are not set to 1 by software. These bits are automatically cleared to 0 when ICRAn is read by the A-DMAC. Accesses by the CPU, AUD, and DMAC have no effect. Each bit is cleared by writing 0 to it after reading as 1. Writing 0 before reading it as 1 has no effect. 0: No input capture has occurred [Clearing conditions] • When writing 0 to this bit after reading it as 1 • When ICRAn is read by A-DMAC 1: Input capture has occurred [Setting condition] • When TCNTA is transferred to ICRAn on assertion of the input capture signal (TIA0n)
4	ICFA4	0	R/(W)*	
3	ICFA3	0	R/(W)*	
2	ICFA2	0	R/(W)*	
1	ICFA1	0	R/(W)*	
0	ICFA0	0	R/(W)*	

Note: n = 0 to 5

* Only writing 0 to this bit after reading it as 1 to clear the flag. Writing 1 to this bit has no effect.

13.10.6 Timer Interrupt Enable Register A (TIERA)

TIERA is an 8-bit readable/writable register that enables and disables an interrupt request of overflow on free-running counter A (TCNTA) or input capture on input capture registers A0 to A5 (ICRA0 to ICRA5). The input capture interrupt request can be used to request DMA transfer between input capture registers and the on-chip RAM according to the settings in the A-DMAC.

TIERA can be read from and written to in byte or word units.

TIERA is initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	OVEA	-	ICEA5	ICEA4	ICEA3	ICEA2	ICEA1	ICEA0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	OVEA	0	R/W	Overflow Interrupt A Enable Enables and disables an OVFA interrupt request when overflow flag A (OVFA) in timer status register A (TSRA) is set to 1. 0: Disables an OVFA interrupt request 1: Enables an OVFA interrupt request
6	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
5	ICEA5	0	R/W	Input Capture Interrupt An Enable Enables and disables an ICFAn interrupt request when bit ICFAn in TSRA is set to 1. When the interrupt is enabled, it can be used to request DMAC transfer by the A-DMAC. 0: Disables a request of input capture interrupt An 1: Enables a request of input capture interrupt An
4	ICEA4	0	R/W	
3	ICEA3	0	R/W	
2	ICEA2	0	R/W	
1	ICEA1	0	R/W	
0	ICEA0	0	R/W	

Note: n = 0 to 5

13.10.7 Input Capture Registers A0 to A5 (ICRA0 to ICRA5)

ICRA0 to ICRA5 are 32-bit read-only registers used for input capturing. Writing to these registers is prohibited.

These registers hold the contents of free-running counter A (TCNTA) when an assertion of input capture signals (TIA00 to TIA05) is detected. At this time, the ICFA bit in timer status register A (TSRA) is set to 1. The bit is cleared to 0 when ICRA0 to ICRA5 are read by the A-DMAC.

The edge to be detected is selected by bit I/O control An in timer I/O control register1A (TIOR1A).

ICRA0 to ICRA5 can be read from in longword units.

ICRA0 to ICRA5 are initialized to H'0000 0000 by a power-on reset or a transition to the hardware standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ICAn[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ICAn[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note: n = 0 to 5

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	ICAn[31:0]	All 0	R	Input Capture An These bits hold 32-bit input capture value.

13.10.8 Free-Running Counter A (TCNTA)

TCNTA is a 32-bit readable/writable register that counts on the signal output by the prescaler via the clock bus, externally input clock signal, or multiplied-and-corrected signal output by timer B.

Timer A is started for counting up by setting the TAE bit in the ATU-III master enable register (ATUENR) to 1. The clock input to the counter is selected by setting the clock select bit (CKSELA) in timer control register A (TCRA).

When TCNTA overflows (from H'FFFF FFFF to H'0000 0000), the overflow flag A (OVFA) in timer status register A (TSRA) is set to 1.

TCNTA can be read from and written to in longword units.

TCNTA is initialized to H'0000 0000 by a power-on reset or a transition to the hardware standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CNTA[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CNTA[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	CNTA [31:0]	All 0	R/W	Timer Count A These bits hold the counter value in 32-bit units.

13.10.9 Noise Canceler Counters A0 to A5 (NCNTA0 to NCNTA5)

NCNTA0 to NCNTA5 are an 8-bit readable/writable registers that are started for counting up by an assertion of externally input signals (TIA00 to TIA05) as a trigger when the noise cancelers are enabled by setting the noise canceler enable bit (NCEA5 to NCEA0) in timer I/O control register 2A (TIOR2A) to 1. These counters are driven by the clock selected in the noise canceler clock select bits (NCKA5 to NCKA0). The counter clock for noise cancelers or clock-bus line 5 are available.

Input edges are processed in premature-transition cancellation or minimum time-at-level cancellation mode depending on the setting of the NCMA bit in the noise cancellation mode register (NCMR) of the common controller, or the NCM1A0 to NCM1A5 bits in the NCMCR1A register of timer A.

- Premature-Transition Cancellation Mode

  When a level change of the externally input signal (TIA00 to TIA05) is detected while bits NCEA5 to NCEA0 are set to 1 and NCNTA0 to NCNTA5 are stopped, NCNTA0 to NCNTA5 are started for counting up. These counters are cleared to H'00 and stopped on the first edge of the Pϕ clock after the value in NCNTA0 to NCNTA5 matches the value in noise canceler registers A0 to A5 (NCRA0 to NCRA5).

  NCNTA0 to NCNTA5 are incremented regardless of the TAE bits in ATUENR.

  The first change is output as the signal whose noise is removed and the edge is to be extracted. Subsequent level changes are masked until the value in the counter reaches the value in the noise canceler register (NCRA0 to NCRA5). The level of the externally input signal is output on this compare match.

  When the NCEA bits are cleared to 0 while the counter (NCNTA0 to NCNTA5) is being incremented, counting continues until the values in the counter and the noise canceler register match. The subsequent level changes are also masked over this period.

- Minimum Time-at-Level Cancellation Mode

  When a level change of the externally input signal (TIA00 to TIA05) is detected while bits NCEA5 to NCEA0 are set to 1 and NCNTA0 to NCNTA5 are stopped, NCNTA0 to NCNTA5 are started for counting up. These counters are cleared to H'00 and stopped on the first edge of the Pϕ clock after the value in NCNTA0 to NCNTA5 matches the value in noise canceler registers A0 to A5 (NCRA0 to NCRA5) or after the level of the externally input signal (TIA00 to TIA05) is changed.

  NCNTA0 to NCNTA5 are incremented regardless of the TAE bits in ATUENR.

When a level change of the externally input signal is detected, the corresponding noise canceler counter (NCNTA0 to NCNTA5) is started for counting up. If subsequent level changes are not detected until the value in the counter reaches the value in the noise canceler register (NCRA0 to NCRA5), the previously accepted level change is output as the signal whose noise is removed on compare match of the counter and noise canceler register. When the subsequent level change is detected before the values in the counter and noise canceler register match, all the changes are treated as noise. Therefore the signal whose noise is removed is not changed.

When the NCEA bits are cleared to 0 while the counter (NCNTA0 to NCNTA5) is being incremented, counting continues until the values in the counter and the noise canceler register match or a level change on the externally input signal is detected.

NCNTA5 to NCNTA0 can be read from and written to in byte or word units.

NCNTA5 to NCNTA0 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	NCCNTAn[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: n = 0 to 5

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	NCCNTAn [7:0]	All 0	R/W	Noise Canceler Count An These bits hold the counter value in byte units.

13.10.10 Noise Canceler Registers A0 to A5 (NCRA0 to NCRA5)

NCRA0 to NCRA5 are 8-bit readable/writable registers that set the upper limitations of noise canceler counters (NCNTA0 to NCNTA5). For example, when the noise canceler is driven by the Pϕ clock divided by 128 and these registers are set to H'FF, a pulse whose width is 1.64 ms (Pϕ is 20 MHz) can be treated as noise.

Input edges are processed in premature-transition cancellation or minimum time-at-level cancellation mode depending on the setting in the NCMA bit of the noise cancellation mode register (NCMR) of the common controller, or the NCM1A0 to NCM1A5 bits in NCMCR1A of timer A.

For details on these modes, see section 13.10.9, Noise Canceler Counters A0 to A5 (NCNTA0 to NCNTA5).

NCRA0 to NCRA5 can be read from and written to in byte or word units.

NCRA0 to NCRA5 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	NCTAn[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: n = 0 to 5

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	NCTAn [7:0]	All 0	R/W	Noise Cancellation Time An These bits set a period for noise cancellation in byte units.

13.11 Operations of Timer A

13.11.1 Operation of Noise Canceler

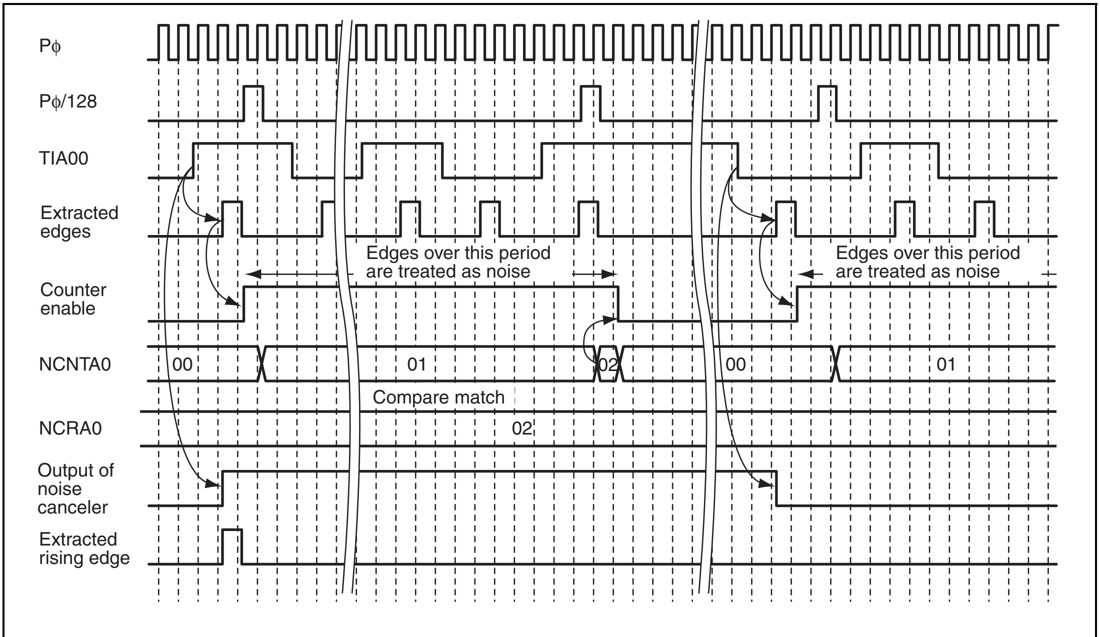
Input edges are processed in premature-transition cancellation or minimum time-at-level cancellation mode depending on the setting in the NCMA bit in the noise cancellation mode register (NCMR) of the common controller, or noise cancellation mode channel register 1A (NCMCR1A) of timer A.

Figures 13.6 and 13.8 show examples of noise cancellation in premature-transition cancellation and minimum time-at-level cancellation modes, respectively. In these examples, edges are input on pin TIA00 and the rising edge sensing is selected.

In premature-transition cancellation mode, noise canceler counter A (NCNTA) is started by the level change of the externally input signal as a trigger. At the same time, the level change is output as the signal after noise removal.

Counting continues until the counter value match the value in noise canceler register A (NCRA). Level changes within the period are ignored and are not output. When values in the counter and NCRA match, the level of the externally input signal is output. Note that the levels are changed on the compare match when the input levels at the start of counting (after the first level change) and on the compare match differ.

Figure 13.7 shows an example of noise cancellation for two types of waveforms.



Note: When the rising-edge sensing and subsequent edge canceled mode are selected

Figure 13.6 Example of Noise Cancellation in Premature-Transition Cancellation

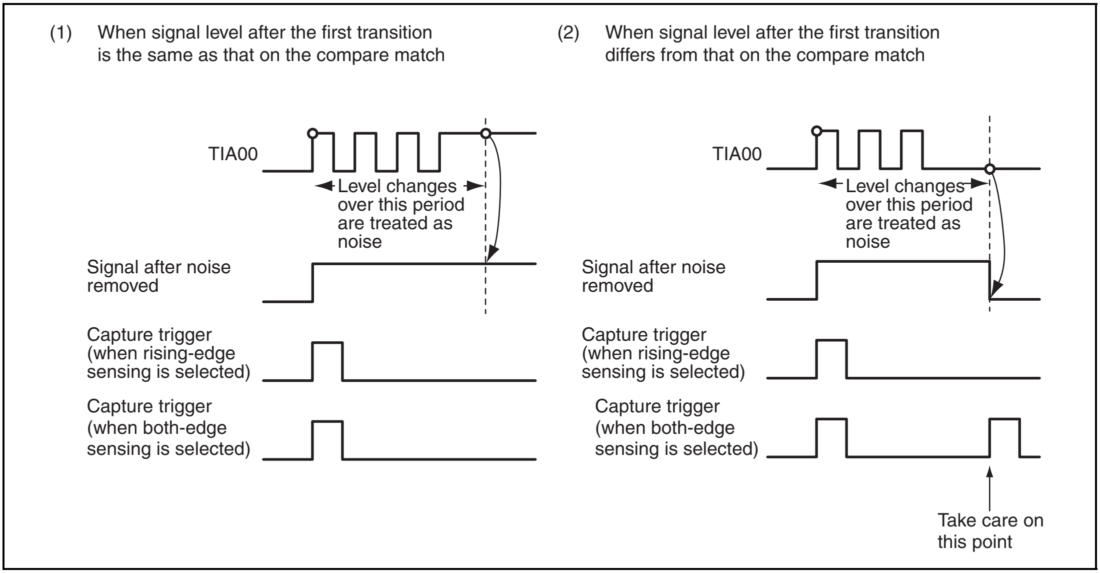
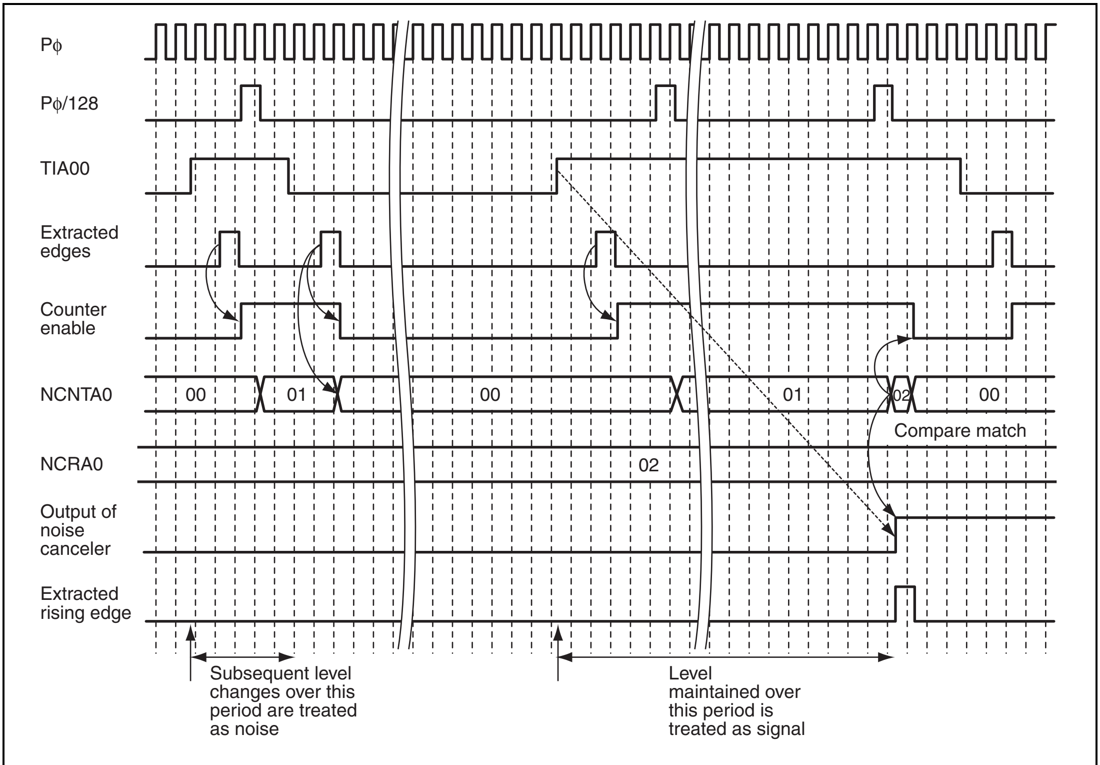

(1) When signal level after the first transition is the same as that on the compare match
TIA00
Level changes over this period are treated as noise
Signal after noise removed
Capture trigger (when rising-edge sensing is selected)
Capture trigger (when both-edge sensing is selected)
(2) When signal level after the first transition differs from that on the compare match
TIA00
Level changes over this period are treated as noise
Signal after noise removed
Capture trigger (when rising-edge sensing is selected)
Capture trigger (when both-edge sensing is selected)
Take care on this point

Figure 13.7 Example of Noise Cancellation in Premature-Transition Cancellation for Two Types of Waveforms

In minimum time-at-level cancellation, noise canceler counter (NCNTA) is started by the level change of the externally input signal as a trigger. Counting continues until the counter value match the value in noise canceler register A (NCRA) or the level change of the input signal is detected.

When the values in the counter and noise canceler register A (NCRA) match, the level change at the start of counting is output as the signal after noise removal. When the level change is detected before the compare match, all the changes are treated as noise. Therefore the signal whose noise is removed is not changed.



Note: Rising-edge sensing and minimum time-at-level cancellation mode are selected in this example.

Figure 13.8 Example of Noise Cancellation in Minimum Time-at-Level Cancellation

13.11.2 Operation of Free-Running Counter

Free-running counter A (TCNTA) is started for counting up by setting the TAE bit in ATU-III master enable register (ATUENR) to 1. When TCNTA overflows (from H'FFFFFFFF to H'00000000), the OVFA bit in timer status register A (TSRA) is set to 1. An interrupt request is issued for the CPU when the OVEA bit in timer interrupt enable register A (TIERA) is set to 1. After overflow, TCNTA continues counting up from H'00000000.

When the TAE bit in AUTENR is cleared to 0, TCNTA is stopped but is not cleared. By setting the TAE bit to 1 again, TCNTA is resumed from the value when stopped.

TCNTA can be written during operation and writing takes priority over counting. After that, TCNTA is started from the written value. Regardless of the clock driving the counter, the write access is completed in two cycles of the Pϕ clock.

The prescalers run regardless of the TAE bit and are not synchronized with the timing at which the TAE bit is set. Therefore, the time from when the TAE bit is set to when TCNTA is incremented for the first time is less than the cycle of the clock of TCNTA.

Figure 13.9 shows an example of free-running counter A (TCNTA) operation.

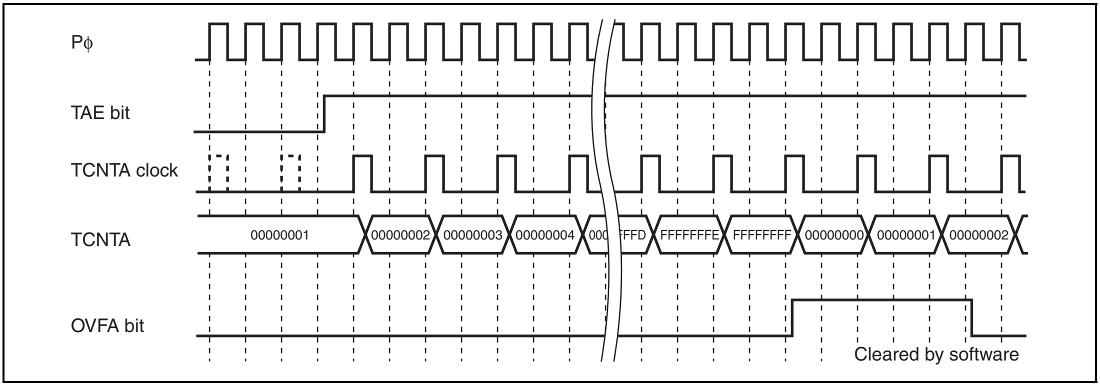


Figure 13.9 Example of Free-Running Counter A (TCNTA) Operation: Overflow Timing

13.11.3 Input Capture

Input capture is performed by input capture registers A0 to A5 (ICRA0 to ICRA5) when input capture is enabled in bits IOA5 to IOA0 in timer I/O control register 1A (TIOR1A). ICRA0 to ICRA5 capture the value in free-running counter A (TCNTA) by an edge of externally input signals (TIA00 to TIA05). Noise on the signals can be removed by the noise cancelers.

TCNTA is started for counting up by setting the TAE bit in AUTENR. When an edge is input on the corresponding signal, the bit in timer status register A (TSRA) is set to 1 and the value in TCNTA is transferred to ICRA. The rising or falling edge, or both edges can be selected. Interrupt requests can be issued for the CPU by setting interrupt enable register A (TIERA) and DMA transfer by the A-DMAC can be also activated by the interrupt requests.

When input capture registers A0 to A5 (ICRA0 to ICRA5) and free-running counter A (TCNTA) are written simultaneously, ICRA0 to ICRA5 capture the previous value stored in TCNTA.

Figure 13.10 shows an example of input capture when the edges to be sensed are rising edges for TIA00, falling edges for TIA01, and both edges for TIA02.

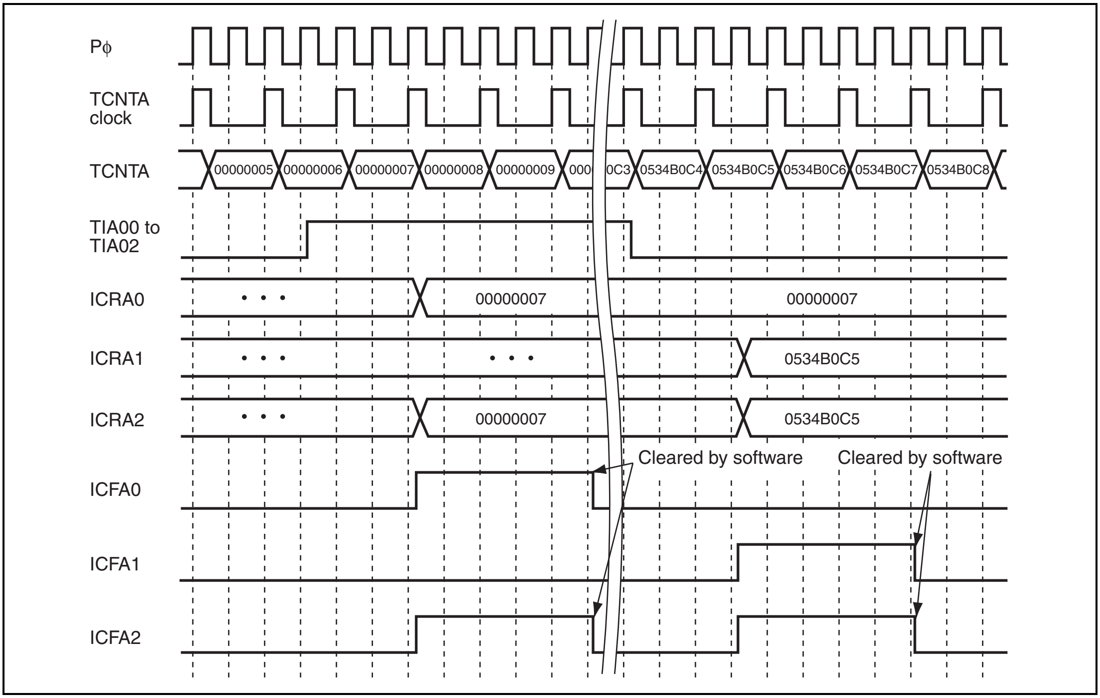
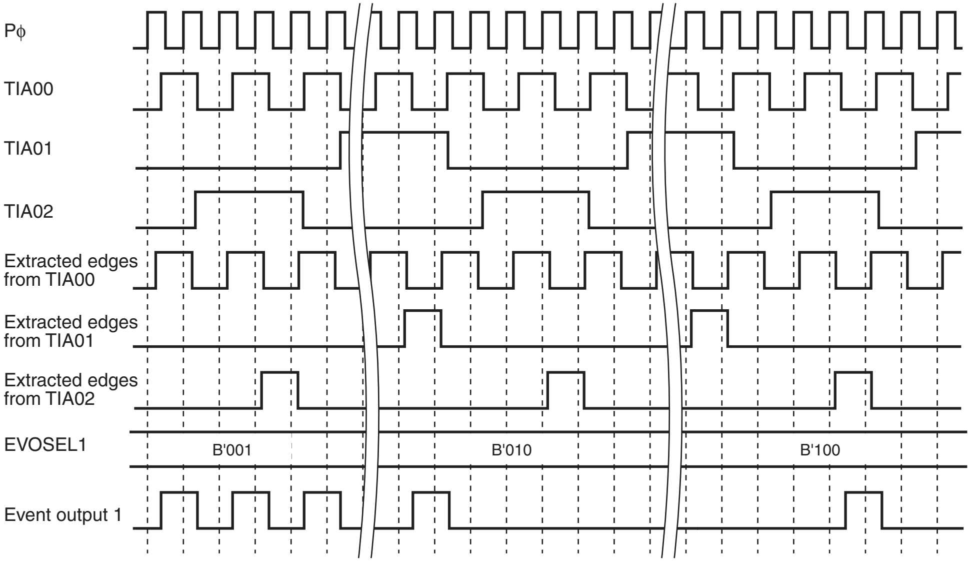


Figure 13.10 Example of Input Capture of Timer A

Signals after noise cancellation and edge extraction can be output to timer B by setting bits EVOSEL1[2:0] in timer control register A (TCRA). By setting bits EVOSEL2A or EVOSEL2B, TIOA01 and TIA02 after noise cancellation and edge extraction can be output to timer D as events.

Figure 13.11 shows an example of event output when TIA00 is selected as event output 1 and TIA01 is selected as event output 1 (EVOSEL1[2:0] = B'001 and B'010, respectively).


Pϕ
TIA00
TIA01
TIA02
Extracted edges from TIA00
Extracted edges from TIA01
Extracted edges from TIA02
EVOSEL1
B'001
B'010
B'100
Event output 1

Note: When the rising-edge sensing is selected for TIA00 to TIA02

Figure 13.11 Example of Event Output

13.11.4 DMA Transfer

The A-DMAC can be activated by an input capture interrupt request. DAM transfer by the A-DMAC transfers data between the input capture register and on-chip RAM. Six 32-bit buffer registers are in the A-DMAC and store data read through DMA transfer. Values previously and currently captured from the input capture register are transferred to the on-chip RAM every interrupt request.

13.12 Overview of Timer B

Timer B generates a multiplied-and-corrected clock signal based on an external-event input and supplies the generated signal to other timers. Timer B consists of an edge-interval measuring block, frequency-multiplied clock generator, and frequency-multiplied clock signal corrector.

(1) Edge-Interval Measuring Block

The edge-interval measuring block measures the intervals between edges of external-event signals input via timer A.

Interrupt requests can be issued for the CPU in response to matches between edge interval measuring counter B0 (TCNTB0) and output compare register B0 (OCRB0). TCNTB0 can also be captured in ICRB2 via ICRB1 on matches between event counter B1 (TCNTB1) and output compare register B1 (OCRB1). Counting by TCNTB1 is driven by the external-event input. This provides a way to measure the intervals between multiple event inputs. Although TCNTB0 is cleared every event input, ICRB1 keeps a running total of the TCNTB0 value. ICRB2 latches the running totals on compare matches of the event counter.

(2) Frequency-Multiplied Clock Generator

The frequency-multiplied clock generator generates a clock signal by producing from 1 to 4095 cycles in response to an external-event input signal.

A down counter is decremented from values captured from edge-interval measuring counter B0 (TCNTB0) in the edge-interval measuring block. The step size for decrementation is the frequency-multiplication ratio. When the down counter underflows, the multiplied clock (AGCK1) signal is asserted. Note that the clock signal generated by this block is only used within timer B and is not output beyond timer B.

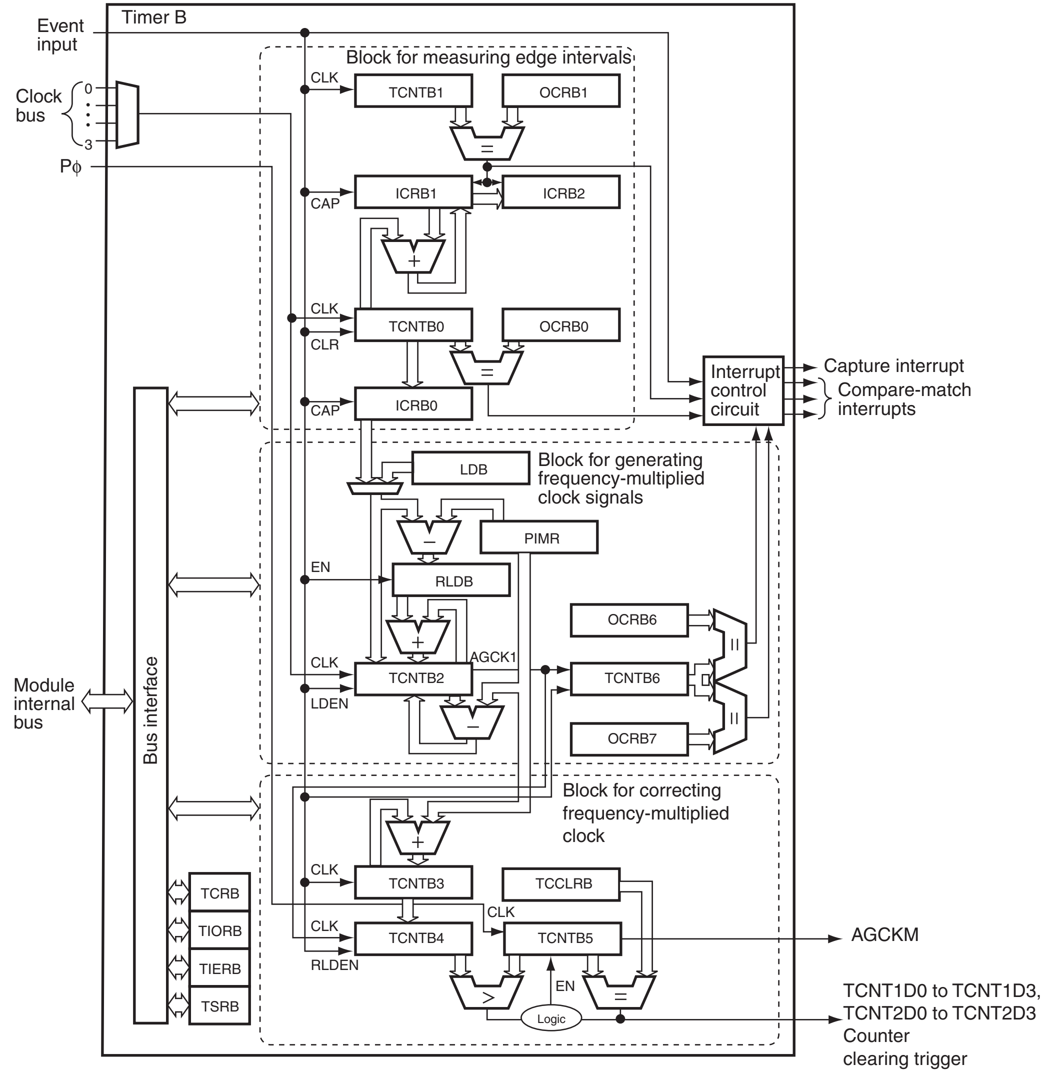
An interrupt request for the CPU can be generated on matches between TCNTB6 and OCRB6. TCNTB6 counts cycles of the frequency-multiplied clock signal.

(3) Frequency-Multiplied Clock Signal Corrector

The multiplied clock signal (AGCK1) needs to be corrected when two consecutive edge intervals differ significantly, since the earlier interval is referred to in calculating the multiplier for the current interval. The frequency-multiplied clock signal corrector generates a multiplied-and-corrected clock signal (AGCKM) by using three correcting counters (TCNTB3 to TCNTB5) and correcting counter clearing register (TCCLRB). Output of the clock signal thus produced on clock-bus line 5 can be selected by using the clock-bus control register (CBCNT). Other timers can then use this clock as a source for counting.

13.12.1 Block Diagram of Timer B

Figure 13.12 is a block diagram of timer B.



[Legend]

TCRB:	Timer control register B	TCNTB5:	Generating frequency-multiplied counter B5
TIORB:	Timer I/O control register B	TCNTB6:	Frequency-multiplied clock counter B6
TIERB:	Timer interrupt enable register B	OCRB:	Output compare register B
TSRB:	Timer status register B	ICRB:	Input capture register B
TCNTB0:	Edge interval measuring counter B0	LDB:	Load register B
TCNTB1:	Event counter B1	PIMR:	Pulse interval multiplier register
TCNTB2:	Reloadable counter B2	RLDB:	Reload register B
TCNTB3:	Correcting event counter B3	TCCLRB:	Correcting counter clearing register B
TCNTB4:	Correcting frequency-multiplied counter B4		

Figure 13.12 Block Diagram of Timer B

13.13 Descriptions of Timer B Registers

13.13.1 Timer Control Register B (TCRB)

TCRB is an 8-bit readable/writable register that selects the clock driving edge interval measuring counter B0 (TCNTB0) and reloadable counter B2 (TCNTB2).

TCRB can be read from and written to in byte or word units.

TCRB is initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	CKSELB	
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	CKSELB	00	R/W	Clock Select B These bits select the clock driving TCNTB0 and TCNTB2 from clock-bus lines 0 to 3, which are clock signals divided by prescalers 0 to 3. The counters are incremented on the rising edge of the selected clock. To select the clock, stop timer B operation. 00: Clock-bus line 0 is selected 01: Clock-bus line 1 is selected 10: Clock-bus line 2 is selected 11: Clock-bus line 3 is selected

13.13.2 Timer I/O Control Register B (TIORB)

TIORB is an 8-bit readable/writable register that selects the source of the frequency-multiplied clock and enables and disables the externally input signals, loading data, and correcting frequency-multiplied clock. TIORB also controls multiplied-and-corrected clock generating counter B5 (TCNTB5) and output compare register B6 (OCRB6).

TIORB can be read from and written to in byte or word units.

TIORB is initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	LD SEL	CTC NTB5	EVC NTB	LDEN	CCS	-	-	IOB6
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	LDSEL	0	R/W	Loading Data Select Selects the register to be loaded to reloadable counter B2 (TCNTB2) and to be used for calculating data to be loaded to reload register B (RLDB) from ICRB0 or LDB. 0: ICRB0 is selected 1: LDB is selected
6	CTCNTB5	0	R/W	Count Control B5 Selects whether or not the multiplied-and-corrected clock generating counter B5 (TCNTB5) is stopped. Setting this bit to 1 stops TCNTB5 and the multiplied-and-corrected clock to be output to other timers. Even if the counter is stopped, it is not cleared. Clearing this bit to 0 resumes TCNTB5 and multiplied-and-corrected clock. 0: TCNTB5 is in operation 1: TCNTB5 is stopped

Bit	Bit Name	Initial Value	R/W	Description
5	EVCNTB	0	R/W	Event Control B Disables and enables the externally input events. Clearing this bit to 0 disables the input. Setting this bit to 1 enables the input with which input capture or generating the multiplied-and-corrected clock signal. 0: Disables the externally input events 1: Enables the externally input events
4	LDEN	0	R/W	Load Enable Selects whether or not the values in reloadable counter B2 (TCNTB2) and reload register B (RLDB) are updated on input capture by ICRB0. 0: TCNTB2 and RLDB are updated on ICEB0 input capture 1: TCNTB2 and RLDB are not updated on ICEB0 input capture
3	CCS	0	R/W	Counter Correction Select Selects whether or not multiplied-and-corrected clock counter B4 (TCNTB4) is stopped when TCNTB3 = TCNTB4. 0: TCNTB4 is in operation when TCNTB3 = TCNTB4 1: TCNTB4 is stopped when TCNTB3 = TCNTB4
2, 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	IOB6	0	R/W	I/O Control B6 Enables and disables compare match between TCNTB6 and OCRB6. When this bit is cleared to 0, compare match between TCNTB6 and OCRB6 is disabled. When it is set to 1, compare match between TCNTB6 and OCRB6 is enabled. When bit CMEB6 in timer interrupt enable register B (TIERB) is set to 1, a compare-match interrupt request is issued for the CPU. 0: Compare match between TCNTB6 and OCRB6 is disabled 1: Compare match between TCNTB6 and OCRB6 is enabled

13.13.3 Timer Status Register B (TSRB)

TSRB is an 8-bit readable/writable register indicating that input capture and compare match has occurred. When interrupts are enabled by the corresponding bits in timer interrupt enable register B (TIERB), an interrupt request is issued for the CPU.

TSRB can be read from and written to in byte or word units.

TSRB is initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	CMFB6	CMFB1	ICFB0	CMFB0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only 0 can be written to this bit after it is read as 1 to clear it. Writing 1 is ignored.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	CMFB6	0	R/(W)*	Compare Match Flag B6 Indicates that compare match between output compare register B6 (OCRB6) and TCNTB6 has occurred. When this bit is read as 1, the compare match has occurred. This bit cannot be set to 1 by software. To clear this bit, write 0 to this bit after read it as 1. Otherwise, writing 1 to this bit is ignored. Only when the IREG bits in TIERB are set to B'10, this bit is automatically cleared to 0 on compare match between TCNTB6 and OCRB6. 0: No compare match has occurred [Clearing conditions] • When writing 0 to this bit after reading it as 1 • When compare match between TCNTB6 and OCRB7 occurs while IREG = B'10 1: Compare match has occurred [Setting condition] • When the values in TCNTB6 and OCRB6 match

Bit	Bit Name	Initial Value	R/W	Description
2	CMFB1	0	R/(W)*	Compare Match Flag B1 Indicates that compare match between output compare register B1 (OCRB6) and TCNTB1 has occurred. When this bit is read as 1, the compare match has occurred. This bit cannot be set to 1 by software. To clear this bit, write 0 to this bit after read it as 1. Otherwise, writing 1 to this bit is ignored. 0: No compare match has occurred [Clearing condition] • When writing 0 to this bit after reading it as 1 1: Compare match has occurred [Setting condition] • When the values in TCNTB1 and OCRB1 match
1	ICFB0	0	R/(W)*	Input Capture Flag B0 Indicates that input capture by input capture registerB0 (ICRB0) has occurred. When this bit is read as 1, the input capture has occurred. This bit cannot be set to 1 by software. To clear this bit, write 0 to this bit after read it as 1. Otherwise, writing 1 to this bit is ignored. 0: No input capture has occurred [Clearing condition] • When writing 0 to this bit after reading it as 1 1: Input capture has occurred [Setting condition] • When the value in TCNTB0 is loaded to ICRB0 by an input capture event as a trigger

Bit	Bit Name	Initial Value	R/W	Description
0	CMFB0	0	R/(W)*	Compare Match Flag B0 Indicates that compare match between output compare register B0 (OCRB0) and TCNTB0 has occurred. When this bit is read as 1, the compare match has occurred. This bit cannot be set to 1 by software. To clear this bit, write 0 to this bit after read it as 1. Otherwise, writing 1 to this bit is ignored. 0: No compare match has occurred [Clearing condition] • When writing 0 to this bit after reading it as 1 1: Compare match has occurred [Setting condition] • When the values in TCNTB0 and OCRB0 match

Note: * Only 0 can be written to this bit after it is read as 1 to clear the flag. Writing 1 to this bit is ignored.

13.13.4 Timer Interrupt Enable Register B (TIERB)

TIERB is an 8-bit readable/writable register that enables and disables interrupt requests on input capture and compare match.

TIERB can be read from and written to in byte or word units.

TIERB is initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	-	-	IREG		CMEB6	CMEB1	ICEB0	CMEB0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
5, 4	IREG[1:0]	00	R/W	Interrupt Enable Edge These bits select the timing at which an interrupt request by CMFB6 in TSRB is output. 00: An interrupt request is output when CMFB6 is enabled 01: An interrupt request is output when the first externally input event is detected after CMFB6 is enabled 10: An interrupt request is output when the second externally input event is detected after CMFB6 is enabled. However, if compare match B7 has occurred before the second event is detected, the interrupt request is not output. 11: Setting prohibited
3	CMEB6	0	R/W	Compare Match Interrupt B6 Enable Enables and disables the output of the interrupt request when compare match flag B6 (CMFB6) is set to 1. The timing at which the interrupt request is output is set by the IREG bits. 0: Output of compare match interrupt B6 is disabled 1: Output of compare match interrupt B6 is enabled
2	CMEB1	0	R/W	Compare Match Interrupt B1 Enable Enables and disables the output of the interrupt request when compare match flag B1 (CMFB1) is set to 1. 0: Output of compare match interrupt B1 is disabled 1: Output of compare match interrupt B1 is enabled
1	ICEB0	0	R/W	Input Capture Interrupt B0 Enable Enables and disables the output of the interrupt request when input capture flag B0 (ICFB0) is set to 1. 0: Input capture flag B0 (ICFB0) is disabled 1: Input capture flag B0 (ICFB0) is enabled
0	CMEB0	0	R/W	Compare Match Interrupt B0 Enable Enables and disables the output of the interrupt request when compare match flag B0 (CMFB0) is set to 1. 0: Output of compare match interrupt B0 is disabled 1: Output of compare match interrupt B0 is enabled

13.13.5 Edge Interval Measuring Counter B0 (TCNTB0)

TCNTB0 is a 32-bit readable/writable register that functions as a counter driven by the clock selected in the clock select B bits of timer control register B (TCRB). TCNTB0 is cleared to H'00000001 on input capture by an externally input event.

TCNTB0 is started when the timer B enable bit (TBE) in ATU-III master enable register (ATUENR) is set to 1. Clearing the TBE bit to 0 stops the counting but the counter value is not cleared.

TCNTB0 can be read from and written to in longword units.

TCNTB0 is initialized to H'0000 0001 by a power-on reset or a transition to the hardware standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CNTB0															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CNTB0															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	CNTB0	H'00000001	R/W	Edge Interval Count These bits store 32-bit counter value.

13.13.6 Input Capture Register B0 (ICRB0)

ICRB0 is a 32-bit read-only register that is loaded with the value in TCNTB0 when an externally input event is detected. At this time, bit ICFB0 in timer status register B (TSRB) is set to 1.

On this input capture of ICRB0, TCNTB0 is cleared to H'00000001.

ICRB0 can be read from in longword units.

ICRB0 is initialized to H'0000 0000 by a power-on reset or a transition to the hardware standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ICB0															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ICB0															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	ICB0	All 0	R	Input Capture B0 These bits store 32-bit captured value.

13.13.7 Output Compare Register B0 (OCRB0)

OCRB0 is a 32-bit readable/writable register that is constantly compared with free-running counter B0 (TCNTB0). When they match, bit CMFB0 in timer status register B (TSRB) is set to 1. An interrupt request is issued for the CPU by setting bit CMEB0 in timer interrupt enable register B (TIERB) to 1.

OCRB0 can be read from and written to in longword units.

OCRB0 is initialized to H'FFFFFFFF by a power-on reset or a transition to the hardware standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	OCB0															
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OCB0															
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	OCB0	All 1	R/W	Output Compare B0 These bits store 32-bit data to be compared with TCNTB0.

13.13.8 Event Counter B1 (TCNTB1)

TCNTB1 is an 8-bit readable/writable register that counts the externally input events when the TBE bit in ATU-III master enable register (ATUENR) is set to 1.

TCNTB1 can be read from and written to in byte or word units.

TCNTB1 is initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	CNTB1							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	CNTB1	All 0	R/W	Event Count B1 These bits store 8-bit counter value.

13.13.9 Output Compare Register B1 (OCRB1)

OCRB1 is an 8-bit readable/writable register that is constantly compared with event counter B1 (TCNTB1). Bit CMFB1 in timer status register B (TSRB) is set to 1 on the first edge of the Pϕ clock after the values in this register and TCNTB1.

An interrupt request is issued for the CPU on compare match by setting bit CMEB1 in timer interrupt enable register B (TIERB).

On this compare match, the value in input capture registerB1 (ICRB1) is loaded to input capture registerB2 (ICRB2) and ICRB1 is cleared.

OCRB1 can be read from and written to in byte or word units.

OCRB1 is initialized to H'FF by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	OCB1							
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	OCB1	All 1	R/W	Output Compare B1 These bits store 8-bit data to be compared.

13.13.10 Input Capture Register B1 (ICRB1)

ICRB1 is a 32-bit read-only register. ICRB1 keeps a running total of the value in edge interval measuring counter B0 (TCNTB0) when the externally input event is detected. ICRB1 is cleared on compare match between event counter B1 (TCNTB1) and output compare register B1 (OCRB1).

ICRB1 can be read from in longword units.

ICRB1 is initialized to H'00000000 by a power-on reset or a transition to the hardware standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ICB1															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ICB1															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	ICB1	All 0	R	Input Capture B1 These bits store 32-bit input capture value.

13.13.11 Input Capture Register B2 (ICRB2)

ICRB1 is a 32-bit read-only register. ICRB1 keeps a running total of the value in input capture registerB1 (ICRB1) on compare match between event counter B1 (TCNTB1) and output compare register B1 (OCRB1).

ICRB2 can be read from in longword units.

ICRB2 is initialized to H'00000000 by a power-on reset or a transition to the hardware standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ICB2															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ICB2															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	ICB2	All 0	R	Input Capture B2 These bits store 32-bit input capture value.

13.13.12 Load Register B (LDB)

LDB is a 32-bit readable/writable register that is aligned with a longword boundary. The lower 24 bits are available.

When the LDSEL bit in timer I/O control register B (TIORB) is set to 1, the value in this register is loaded to reloadable counter B2 (TCNTB2) and reload register B (RLDB).

LDB can be read from and written to in longword units.

LDB is initialized to H'00000000 by a power-on reset or a transition to the hardware standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	LDVAL							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LDVAL															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 0	LDVAL	All 0	R/W	Load Value These bits store 24-bit data to be loaded to TCNTB2 and RLDB.

13.13.13 Reload Register B (RLDB)

RLDB is a 32-bit readable/writable register that is aligned with a longword boundary. The upper 24 bits are available.

This register is updated when the externally input event is detected while the LDEN bit in timer I/O control register B (TIORB) is cleared to 0.

The value in input capture registerB0 (ICRB0) or load register B (LDB) minus the value in the PIMR bits in PIM is used for updating. ICRB0 or LDB is set by the LDSEL bit in TIORB. For subtraction on ICRB0 and PIMR, the lower 24 bits of ICRB0 and PIM which is zero-extended.

The value in this register is added to the value in TCNTB2 on the first counter clock after the value in bits reload count B2 (CNTB2) is equal to or less than the value in the pulse interval multiplier register (PIM).

RLDB can be read from and written to in longword units.

RLDB is initialized to H'00000000 by a power-on reset or a transition to the hardware standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RLDVAL															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RLDVAL								-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	RLDVAL	All 0	R/W	Reload Value These bits store 24-bit reload value.
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

13.13.14 Reloadable Counter B2 (TCNTB2)

TCNTB2 is a 32-bit readable/writable register that functions as a counter driven by the clock selected in bit clock select B (CKSELB) in timer control register B (TCRB). Each decrementation is by the value set in the pulse interval multiplier register (PIMR).

When the TBE bit in ATU-III master enable register (ATUENR) is set to 1, this counter is decremented. Even if the TBE bit is cleared to 0, this counter is not cleared.

This counter is updated when the externally input event is detected while the LEDN bit in timer I/O control register B (TIORB) is cleared to 0.

The value in input capture registerB0 (ICRB0) or load register B (LDB) is used for updating. ICRB0 or LDB is set by the LDSEL bit in TIORB.

The value in reload register B (RLDB) is added to the value stored in this counter on the first counter clock after the value in this counter is equal to or less than the value in PIM. A single pulse whose width is equal to the cycle of the Pϕ clock on reloading. The pulse is the frequency-multiplied clock (AGCK1).

TCNTB2 can be read from and written to in longword units.

TCNTB2 is initialized to H'00000000 by a power-on reset or a transition to the hardware standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CNTB2															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CNTB2								-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	CNTB2	All 0	R/W	Reload Count B2 These bits store 24-bit reload counter value.
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

13.13.15 Pulse Interval Multiplier Register (PIMR)

PIMR is a 16-bit readable/writable register that sets the multiplication ratio of the externally input event for generation of the frequency-multiplied clock.

The settable value ranges from 1 (H'001) to 4095 (H'FFF). Do not set the PIM bits to H'000. If the PIM bits are set to H'000, operation cannot be guaranteed.

The value in this register is used in various registers; the step size in decrementation of reloadable counter B2 (TCNTB2); calculation of the value to be input to reload register B (RLDB); calculation of the value to be input to corrected event counter B3 (TCNTB3).

PIMR can be read from and written to in word units.

PIMR is initialized to H'0001 by a power-on reset or a transition to the hardware standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	PIM											
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 0	PIM	H'001	R/W	Pulse Interval Multiplier These bits set the multiplication ratio for the frequency-multiplied clock. The settable value ranges from 1 to 4095.

13.13.16 Multiplied Clock Counter B6 (TCNTB6)

TCNTB6 is a 32-bit readable/writable register that functions as an up-counter driven by the frequency-multiplied clock (AGCK1). This counter is cleared to H'00000000 when the external input event is detected.

When the TBE bit in ATU-III master enable register (ATUENR) is set to 1, this counter is incremented. Even if the TBE bit is cleared to 0 and counting up is stopped, the counter value is not cleared.

TCNTB6 can be read from and written to in longword units.

TCNTB6 is initialized to H'0000 0000 by a power-on reset or a transition to the hardware standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CNTB6															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CNTB6				-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	CNTB6	All 0	R/W	Frequency-Multiplied Clock Count B6 These bit store 20-bit counter value driven by the frequency-multiplied clock
11 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

13.13.17 Output Compare Register B6 (OCRB6)

OCRB6 is a 32-bit readable/writable register that is aligned with a longword boundary. The upper 20 bits are available.

This register enables and disables compare match between multiplied clock counter B6 (TCNTB6) and OCRB6 when bit IOB6 in timer I/O control register B (TIORB) is set.

When the edge of the frequency-multiplied clock (AGCK1) is input while the values in this counter and TCNTB6 have matched with compare match enabled by bit IOB6, bit CMFB6 in timer status register B (TSRB) is set to 1. An interrupt request is issued for the CPU by setting bit CMEB6 in timer interrupt enable register B (TIERB) to 1.

OCRB6 can be read from and written to in longword units.

OCRB6 is initialized to H'FFFFF000 by a power-on reset or a transition to the hardware standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	OCB6															
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OCB6				-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	OCB6	All 1	R/W	Output Compare B6 These bits store 20-bit data to be compared.
11 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

13.13.18 Output Compare Register B7 (OCRB7)

OCRB7 is a 32-bit readable/writable register that is aligned with a longword boundary. The upper 20 bits are available.

When the edge of the frequency-multiplied clock (AGCK1) is input while the values in this counter and TCNTB6 have matched with the IREG bits in timer interrupt enable register B (TIERB) set to B'10, bit CMFB6 in timer status register B (TSRB) is cleared to 0.

There is no status flag indicating the compare match. An interrupt request cannot be issued for the CPU.

OCRB7 can be read from and written to in longword units.

OCRB7 is initialized to H'FFFFF000 by a power-on reset or a transition to the hardware standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	OCB7															
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OCB7				-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	OCB7	All 1	R/W	Output Compare B7 These bit store 20-bit data to be compared.
11 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

13.13.19 Correcting Event Counter B3 (TCNTB3)

TCNTB3 is a 32-bit readable/writable register.

When the externally input event is detected, the value in this counter is transferred to multiplied-and-corrected clock counter B4 (TCNTB4) and then is incremented by the value in the pulse interval multiplier register (PIMR)

When the TBE bit in ATU-III master enable register (ATUENR) is set to 1, this counter is incremented. Even if the TBE bit is cleared to 0, the counter value is not cleared.

TCNTB3 can be read from and written to in longword units.

TCNTB3 is initialized to H'00000000 by a power-on reset or a transition to the hardware standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CNTB3															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CNTB3				-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	CNTB3	All 0	R/W	Correcting Event Count B3 These bits store 20-bit event count value.
11 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

13.13.20 Multiplied-and-Corrected Clock Counter B4 (TCNTB4)

TCNTB4 is a 32-bit readable/writable register that is a 20-bit up-counter. The value in TCNTB3 is loaded to this counter when the externally input event is detected.

This up-counter is driven by the frequency-multiplied clock (AGCK1) output by reloadable counter B2 (TCNTB2) and is cleared to H'00000 when the externally input event is detected while CNTB3 = H'00000.

When the values in this counter and TCNTB3 match with the CCS bit in timer I/O control register B (TIORB) set to 1, counting is stopped.

When the TBE bit in ATU-III master enable register (ATUENR) is set to 1, this counter is incremented. Even if the TBE bit is cleared to 0 and counting is stopped, the counter value is not cleared.

TCNTB4 can be read from and written to in longword units.

TCNTB4 is initialized to H'0000 0000 by a power-on reset or a transition to the hardware standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CNTB4															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CNTB4				-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	CNTB4	All 0	R/W	Multiplied-and-Corrected Clock Count B4 These bits store 20-bit frequency-multiplied clock count value.
11 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

13.13.21 Multiplied-and-Corrected Clock Generating Counter B5 (TCNTB5)

TCNTB5 is a 32-bit readable/writable register.

This counter is enabled by the TBE bit in ATU-III master enable register (ATUENR) and the count control B5 bit (CTCNTB5). Incrementation of this counter is driven by the Pϕ clock as long as the counter value is less than the value in multiplied-and-corrected clock counter B4 (TCNTB4). Incrementation is stopped when the counter value matches the value in correcting counter clearing register B (TCCLRB).

The value in this counter is corrected and cleared when the externally input event is detected while TCNTB3 = H'00000. The value for the clearing depends on the counter value. When the counter value is equal to the value in TCCLRB, this counter is cleared to H'00001000. When the counter value is not equal to the value in TCCLRB, incrementation continues until the counter value reaches the value in TCCLRB and then TCNTB5 is cleared to H'00001000.

Every incrementation of this counter, a single pulse of the multiplied-and-corrected clock (AGCKM) is output. The clock can be output on clock-bus line 5 by setting the CB5SEL bit in clock bus control register (CBCNT) to 1. The output of the clock is temporarily stopped by altering the setting in bit CTCNTB5.

TCNTB5 can be read from and written to in longword units.

TCNTB5 is initialized to H'0000 1000 by a power-on reset or a transition to the hardware standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CNTB5															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CNTB5				-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	CNTB5	H'00001	R/W	Multiplied-and-Corrected Clock Generation Count B5 These bit store 20-bit multiplied-and-corrected clock count value.
11 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

13.13.22 Correcting Counter Clearing Register B (TCCLRB)

TCCLRB is a 32-bit readable/writable register.

TCCLRB is constantly compared with TCNTB5. When they match, TCNTB5 is stopped and a counter clearing trigger is output to timer D. TCNT1Dn and TCNT2Dn in timer D are separately cleared by setting the corresponding counter clearing enable bit in timer control register Dn (TCRDn).

TCCLRB can be read from and written to in longword units.

TCCLRB is initialized to H'00000000 by a power-on reset or a transition to the hardware standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CCLRB															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CCLRB				-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	CCLRB	All 0	R/W	Correcting Counter Clear B These bits store 20-bit correcting counter clear value.
11 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

13.14 Operations of Timer B

13.14.1 Edge Interval Measuring Function and Edge Input Stopping Function

For timer B, input-capture and compare-match operations are unconditionally performed with input capture register B0 (ICRB0) and output compare register B0 (OCRB0), respectively. These registers are connected to free-running counter B0 (TCNTB0).

Operation of timer B is started by setting the TBE bit in the ATU-III master enable register (ATUENR).

ICRB0 captures the TCNTB0 value when an event (the AGCK signal) is input via timer A. After that, TCNTB0 is set to H'00000001. If the interrupt is enabled by the setting in timer interrupt enable register B (TIERB), an interrupt request is also generated for the CPU. This enables measurement of the interval between external event edges.

The value captured by ICRB0 is transferred to the frequency-multiplied clock signal generator, where it is used to be set in reloadable counter B2 (TCNTB2) and reload register B (RLDB).

A compare-match interrupt when the TCNTB0 value reaches the OCRB0 value can also be requested. This interrupt indicates that active edge input has stopped for at least time equivalent to the setting in OCRB0.

Figure 13.13 shows input-capture and compare-match operation of TCNTB0.

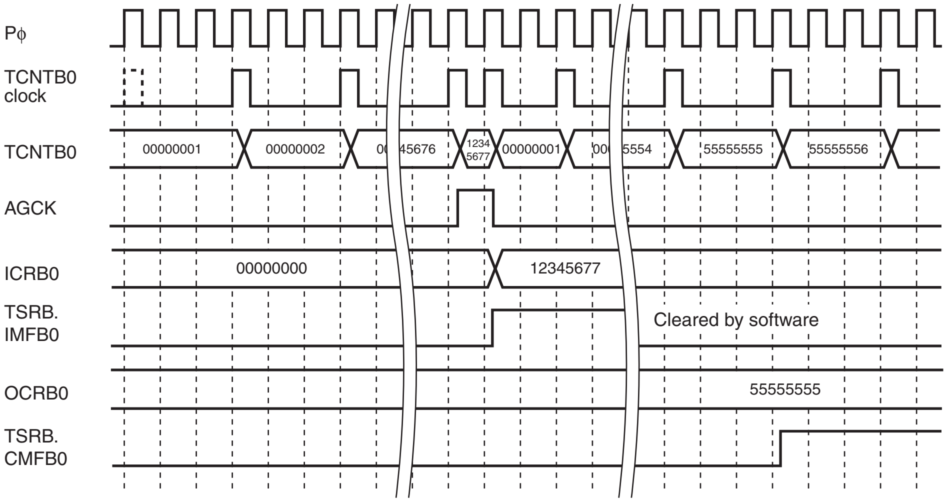


Figure 13.13 Input-Capture and Compare-Match Operation of TCNTB0

Event counter B1 (TCNTB1) counts the input of signals indicating external events (AGCK). When a predetermined value is set in output compare register B1 (OCRB1) and TCNTB1 reaches that value, a compare-match occurs. At this time, input capture register B2 (ICRB2) captures the value in input capture register B1 (ICRB1). An interrupt request for the CPU will also be generated if bit CMEB1 in TIERB is set to 1. The interrupt indicates that the input of edges of the external-event signal has stopped for at least time equivalent to the setting in OCRB1.

The external event signal (AGCK) drives the capturing of TCNTB0 values in ICRB1. Moreover, latching of ICRB1 in ICRB2 on matches between TCNTB1 and OCRB1 can also be selected. This enables the measurement of multiple edge-to-edge intervals.

Figure 13.14 shows compare-match operation of TCNTB1 and capture operation of ICRB1 and ICRB2.

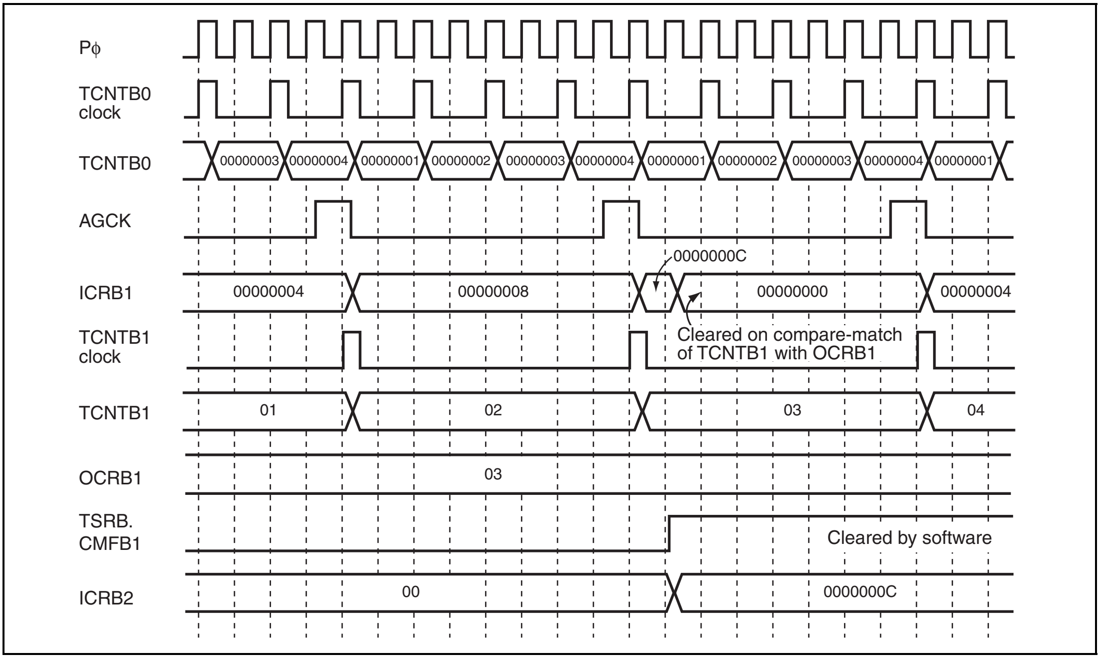


Figure 13.14 Compare-Match Operation of TCNTB1 and Capture Operation of ICRB1 and ICRB2

13.14.2 Frequency-Multiplied Clock Generator

The frequency-multiplied clock generator generates a clock signal (AGCK1) for use within timer B. The cycle of this clock signal is obtained by dividing the intervals between selected transitions of the external-event input (AGCK) by the value in the pulse interval multiplier register (PIMR).

On the selected transition of the external-event input signal, the lower 24 bits of the value captured in ICRB0 of the edge-interval measuring block are transferred to reloadable counter B2 (TCNTB2). At the same time, the value transferred to TCNTB2 minus the value in PIMR (PIM) is transferred to reload register B (RLDB).

When the LDSEL bit in timer I/O control register B (TIORB) is set to 1, the value in load register (LDB) instead of that in ICRB0 can be transferred to TCNTB2 and RLDB.

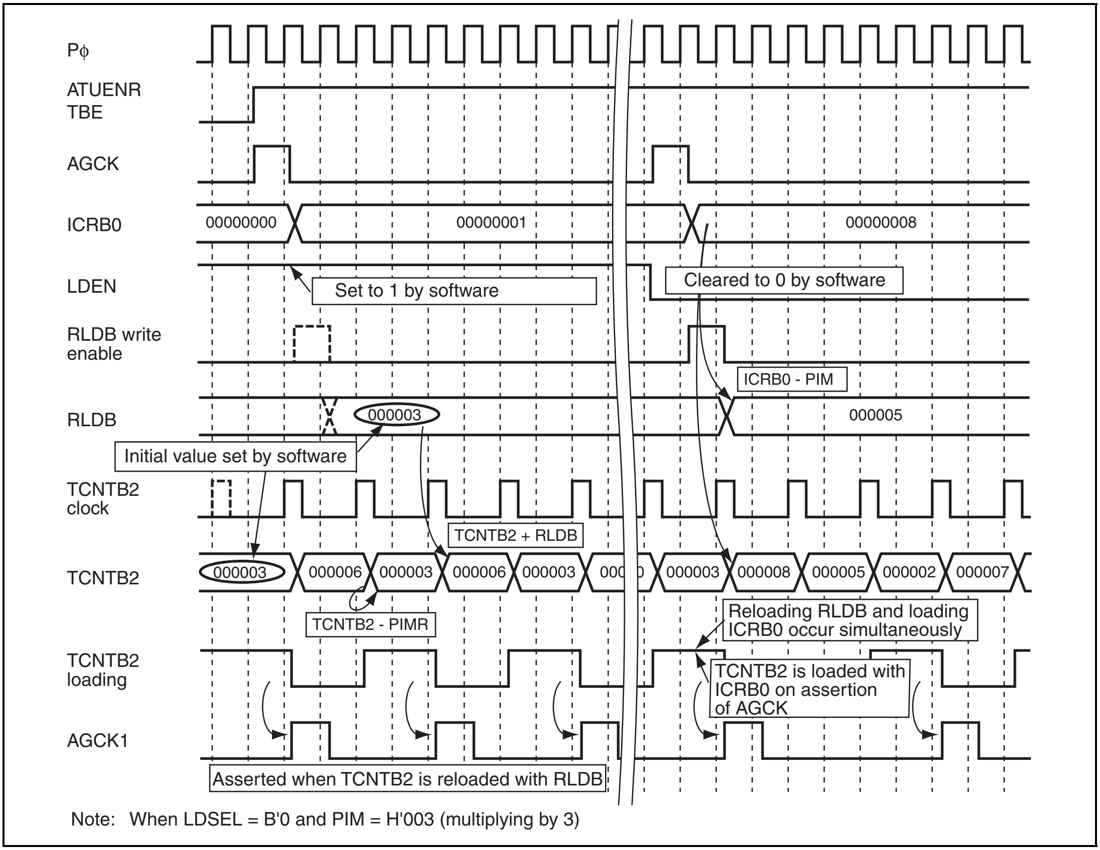
Reloadable counter B2 (TCNTB2) is driven by the clock selected by the CKSELB bits in timer control register B (TCRB). Each decrementation is by the value set in PIMR. When the value in the down counter is less than or equal to that of the PIM bits, RLDB is automatically read out into TCNTB2, which again starts to count down with the same step size (the value in PIMR). A single pulse of the multiplied clock signal (AGCK1) is output in synchronization with the reloading of TCNTB2. The pulse width is equal to the cycle of the Pϕ clock.

Frequency-multiplied clock counter B6 (TCNTB6) is driven by the frequency-multiplied clock signal (AGCK1). Compare match operation can be performed when the values in TCNTB6 and output compare register B6 (OCRB6) match.

Incrementaion of TCNTB6 on the assertion of AGCK1 is unconditional. The values in TCNTB6 and output compare register B6 (OCRB6) are tested for matches, and an interrupt request will be generated for the CPU when the values match, if this interrupt has been enabled in timer interrupt enable register B (TIERB). The IREG bits in TIERB can be set so that the interrupt is generated on the match, on the first AGCK pulse after the match, or on the second AGCK pulse after the match.

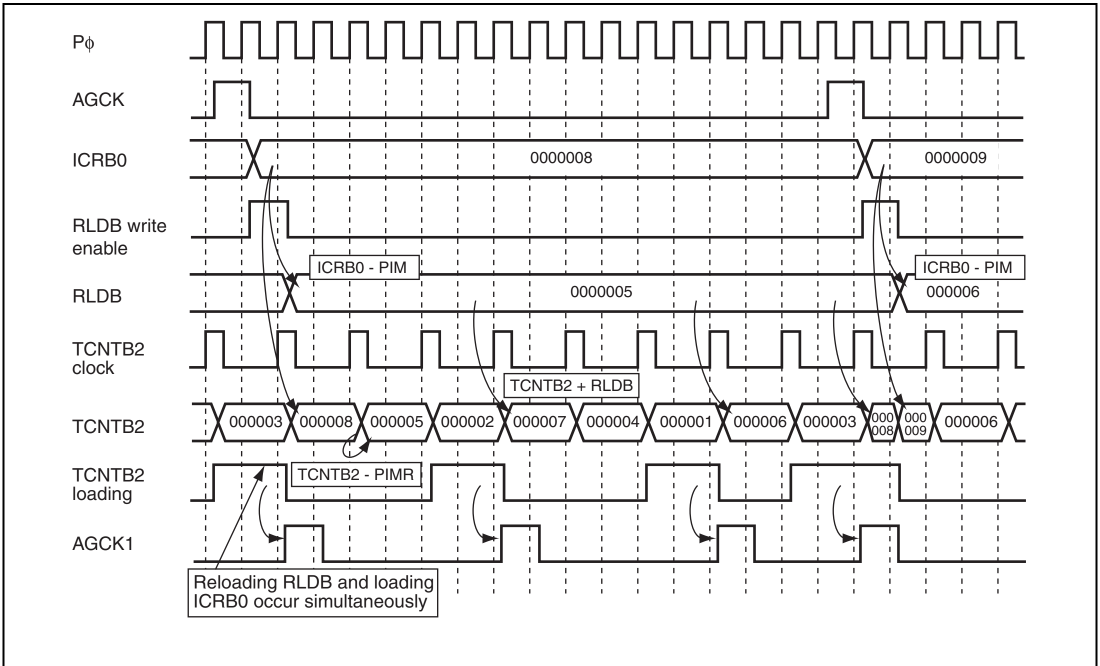
Since AGCK1 is generated with reference to the previous edge-to-edge interval, if two consecutive edge intervals differ significantly, the clock will not be generated correctly. To correct this error, AGCK1 is corrected by the frequency-multiplied clock signal corrector which produces the multiplied-and-corrected clock (AGCKM; described in section 13.14.3, Frequency-Multiplied Clock Signal Corrector).

Figures 13.15 and 13.16 show counting operations with reloading and the frequency-multiplied clock and figures 13.17 and 13.18 show the generation of interrupt requests on matches between TCNTB6 and CMFB6, enabled or disabled by the setting in IREG.



Note: When LDSEL = B'0 and PIM = H'003 (multiplying by 3)

Figure 13.15 Counting Operations with Reloading and Output of Frequency-Multiplied Clock (1)



Note: When TBE = B'1, LDEN = B'0, LDSEL = B'0, and PIM = H'003 (multiplying by 3)

Figure 13.16 Counting Operations with Reloading and Output of Frequency-Multiplied Clock (2)

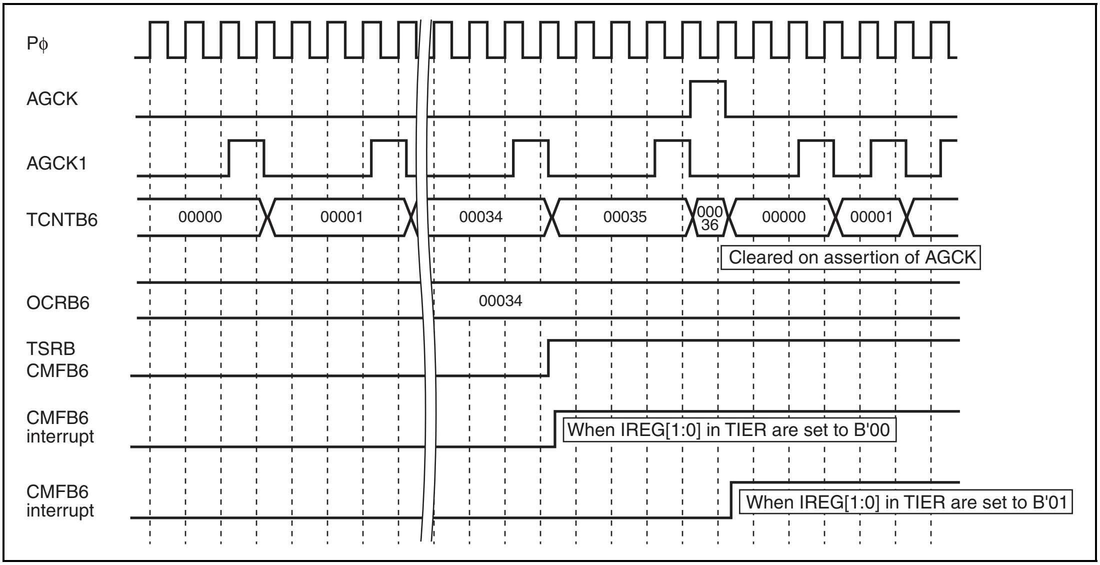

Pφ
AGCK
AGCK1
TCNTB6
00000
00001
00034
00035
000
36
00000
00001
Cleared on assertion of AGCK
OCRB6
00034
TSRB
CMFB6
CMFB6
interrupt
When IREG[1:0] in TIER are set to B'00
CMFB6
interrupt
When IREG[1:0] in TIER are set to B'01

Figure 13.17 Compare-Match Operation of TCNTB6 and Output of CMFB6 Interrupt (IREG = B'00, B'01)

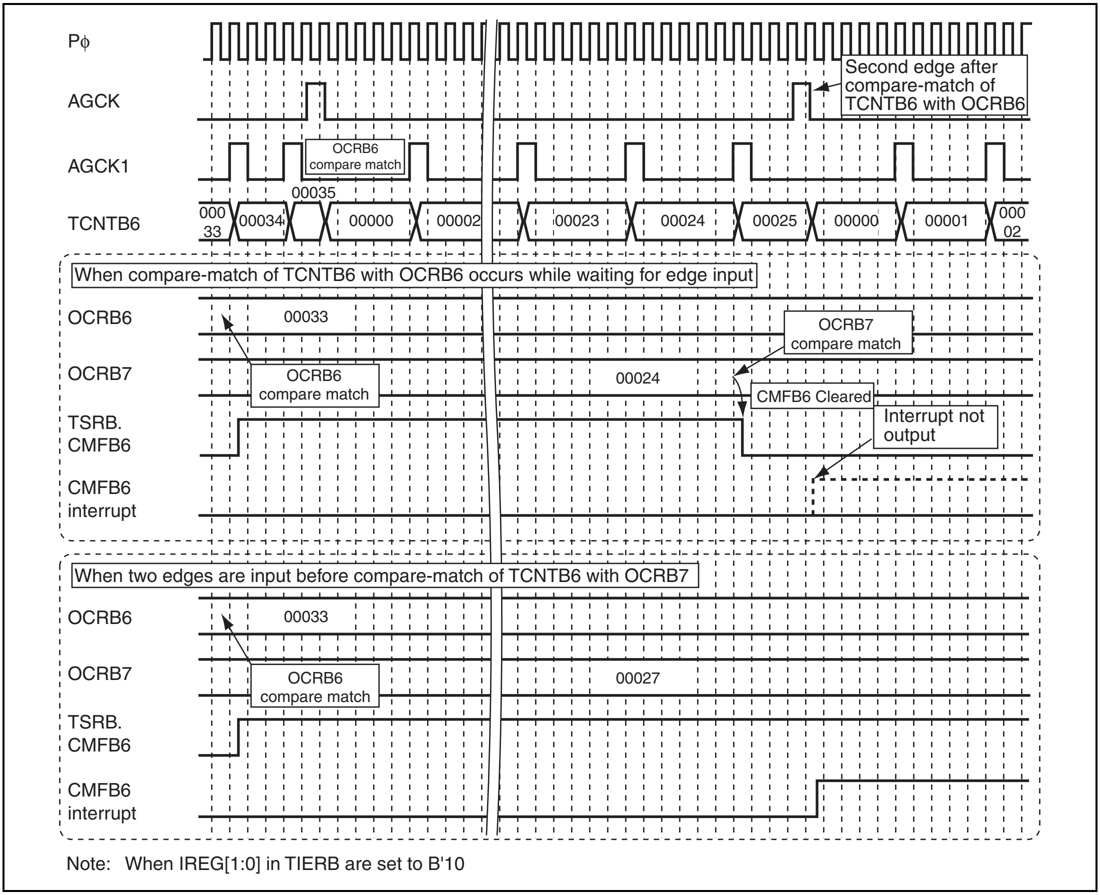


Figure 13.18 CMFB6 Interrupt Output when IREG = B'10

13.14.3 Frequency-Multiplied Clock Signal Corrector

The frequency-multiplied clock signal which is generated by dividing the intervals between external event inputs by the multiplication ratio set in the PIM bits in PIMR can be corrected by using correcting event counter B3 (TCNTB3), multiplied-and-corrected clock counter B4 (TCNTB4), multiplied-and-corrected clock generating counter B5 (TCNTB5), and correcting counter clearing register B (TCCLRB).

TCNTB3 is a 20-bit up-counter that is driven by the external event input (AGCK). On the selected transition of the AGCK signal, the value in TCNTB3 is transferred to TCNTB4, after which TCNTB3 is incremented by the value in the PIM bits.

TCNTB4 is a 20-bit up-counter that is driven by the multiplied clock signal (AGCK1). TCNTB3 is loaded to TCNTB4 with the AGCK input as a trigger, and incrementation of TCNTB4 is driven by the AGCK1 input.

The counter correcting select bit (CCS) in TIORB controls counting by TCNTB4; that is, it selects whether or not counting stops when TCNTB3 = TCNTB4.

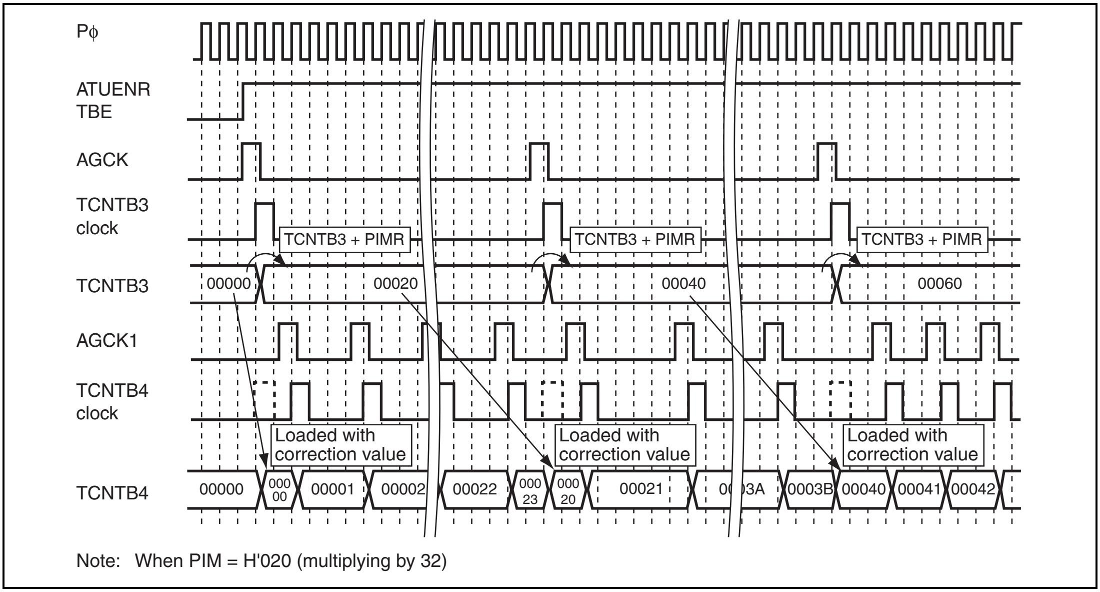
TCNTB5 is a 20-bit up-counter that is driven by the Pϕ clock, meaning that it operates at a high speed. TCNTB5 is constantly compared with TCNTB4 and is incremented as long as its value is lower than that in TCNTB4. Each time TCNTB5 is actually incremented, it produces a single pulse whose width is equal to one cycle of the Pϕ clock. pulse of the peripheral clock signal, namely the multiplied-and -corrected clock signal (AGCKM), for which output on clock-bus line 5 can be selected by bit CB5SEL in the clock bus control register (CBCNT). The AGCKM signal is then available on clock-bus line 5 as a source to drive counting by other timers.

As state above, TCNTB5 is not incremented when its value is greater that in TCNTB4 (for example, after TCNTB3 has been loaded to TCNTB4), TCNTB5 can also be disabled by the count control B5 (CTCNTB5) bit in timer I/O control register B (TIORB). This halts the output of the AGCKM signal.

As long as its value is lower that that in TCNTB4, TCNTB5 is incremented until it reaches the value in correcting counter clearing register B (TCCLRB). Incrementation of TCNTB5 then stops, regardless of the relation between its value and that of TCNTB4. In addition, counters of timer D (TCNT1Dn and TCNT2Dn) can be separately cleared by this match as a trigger when the corresponding counter clearing enable bit (C1CEDn/C2CEDn) in timer control register Dn (TCRDn) is set to 1.

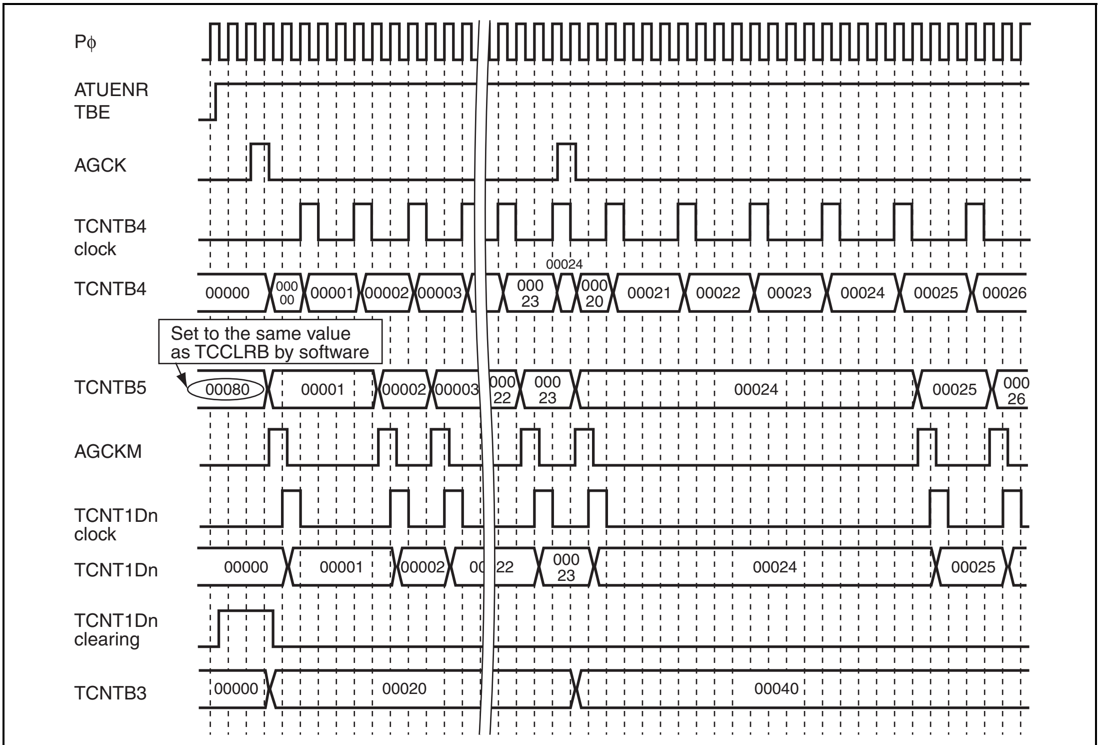
TCNTB4 is unconditionally cleared to H'00000000 when a pulse of the external-event signal (AGCK) is input while TCNTB3 = H'00000000. TCNTB5 is unconditionally set to H'00001000 when a pulse of the external-event signal (AGCK) is input while TCNTB3 = H'00000000. However, when TCNTB5 has not reached TCCLRB, TCNTB5 is incremented until it reaches TCCLRB. After that, it is set to H'00001000.

Figure 13.19 shows operations of TCNTB3 and TCNTB4, figure 13.20 shows operation when TCNTB5 is being started up, figure 13.21 shows TCNTB5 operation with correction at the end of a cycle, and figure 13.22 shows TCNTB5 operation with no correction at the end of a cycle.



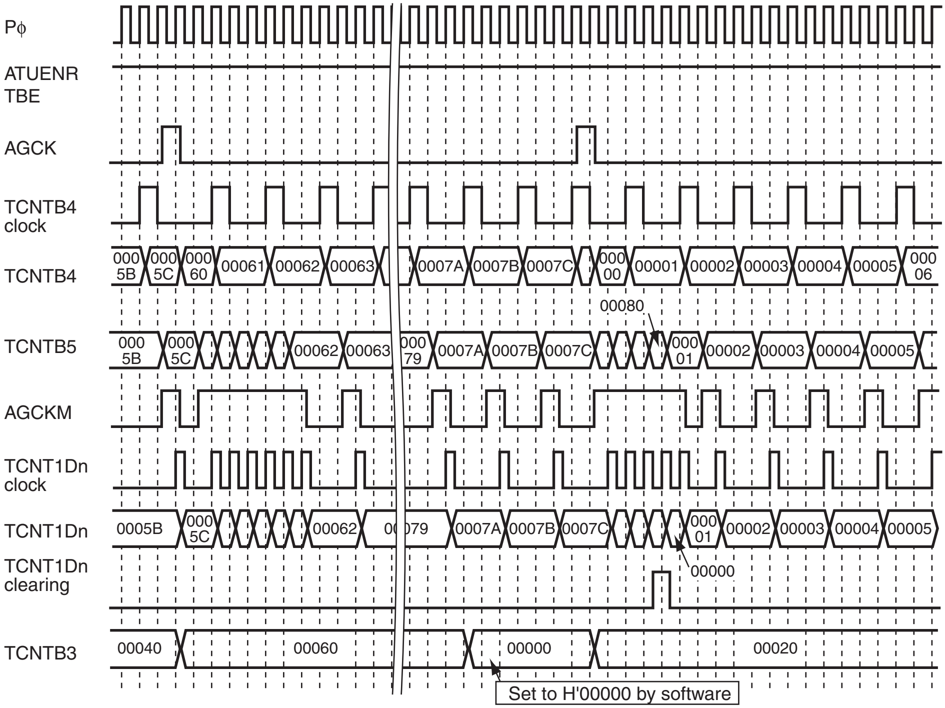
Note: When PIM = H'020 (multiplying by 32)

Figure 13.19 Operation of TCNTB3 and TCNTB4



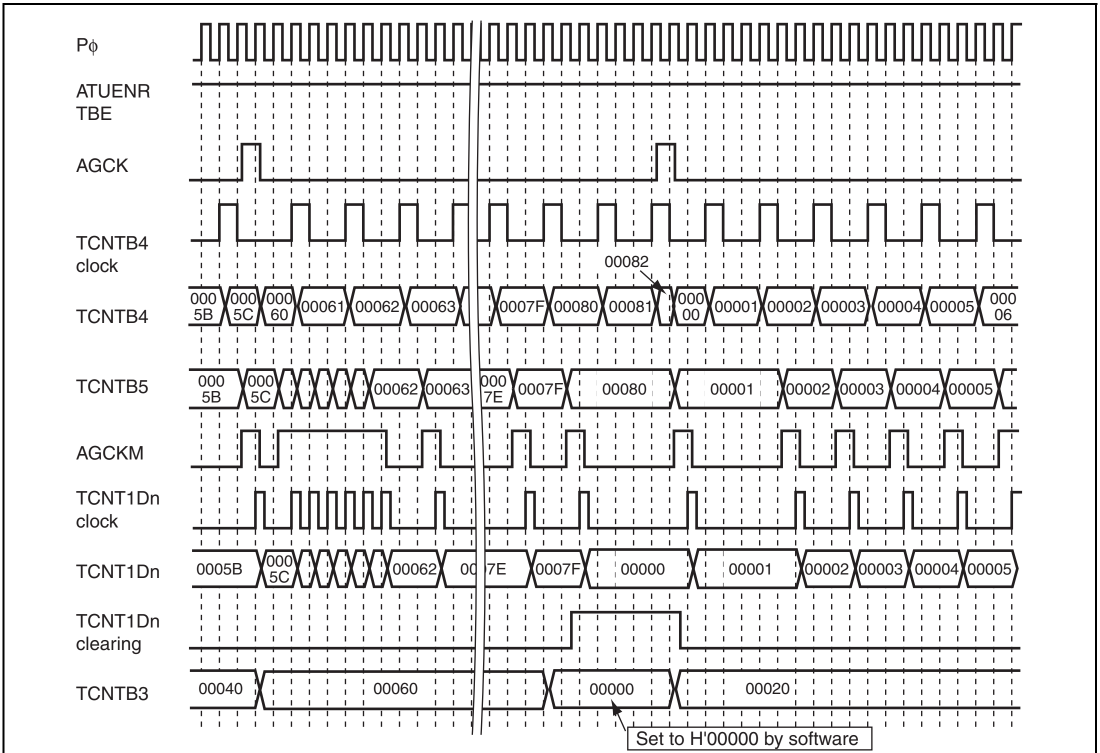
Note: When PIM = H'020 (multiplying by 32) and TCCLRB = H'00080
This figure shows the case of TCNT1Dn. This is the same for TCNT2Dn.

Figure 13.20 TCNTB5 Operation (at Start-Up)



Note: When PIM = H'020 (multiplying by 32) and TCCLRB = H'00080
This figure shows the case of TCNT1Dn. This is the same for TCNT2Dn.

Figure 13.21 Operation of TCNTB5 (with Correction at End of Cycle)



Note: When PIM = H'020 (multiplying by 32) and TCCLRB = H'00080
This figure shows the case of TCNT1Dn. This is the same for TCNT2Dn.

Figure 13.22 Operation of TCNTB5 (without Correction at End of Cycle)

13.15 Overview of Timer C

Timer C consists of five subblocks which has the same functions (listed below).

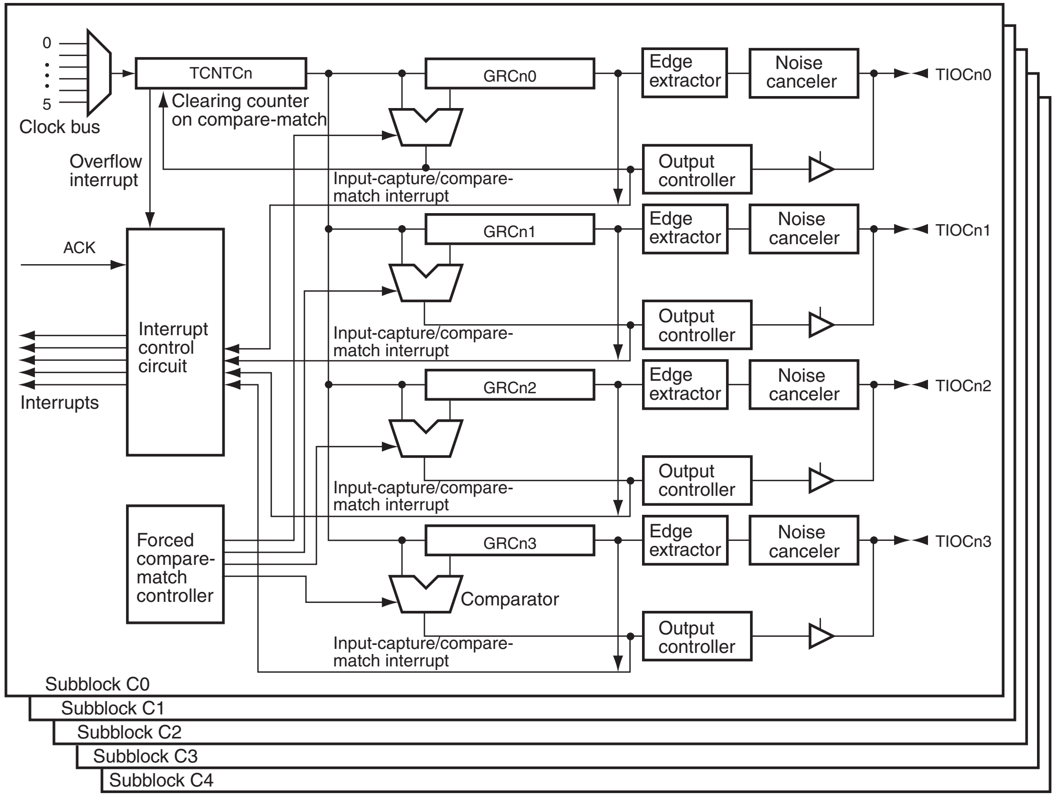
- Input capture and output compare matches
- Choice of rising edge, falling edge, or both edge sensing as the edge of input capture trigger signal
- Output of a waveform on compare match
- Choice of a logical one, a logical zero, or a toggled output by setting a register
- Output of an interrupt request on input capture or compare match
- GRC00, GRC10, GRC20, GRC30, and GRC40, which are used for interrupt requests for the DMAC, can be cleared by the ACK signal.
- Output of an interrupt request on timer counter overflow
- Clearing counter GRCn0 on compare match (not supported by GRCn1 to GRCn3)
- Output of forced compare match by setting the forced compare match bit
- Each input capture trigger input has noise canceling function.

13.15.1 Block Diagram of Timer C

Timer C consists of five subblocks, each of which has the same functions. Each subblock consists of timer counter C (TCNTC), four general registers (GRCm), and controller. The general registers can be used for input capture/compare match operations and input capture trigger input and output compare output signals (TIOCnm) are available.

The initial output value on the TIOCnm pin is 0 for output compare operation. During operation, the previous state is reflected.

Figure 13.23 is a block diagram of timer C.



Note: n = 0 to 4 (n denotes the subblock number)
Only GRCn0 in each subblock can issue interrupt requests for the DMAC.

[Legend]
TCNTC: Timer counter C
GRC: General register C

Figure 13.23 Block Diagram of Timer C

13.16 Description of Timer C Registers

13.16.1 Timer Start Register C (TSTRC)

TSTRC is an 8-bit readable/writable register that enables and disables timer counter Cn (TCNTCn) in subblocks C0 to C4. When the both the STRC bits in this register and the TCE bit in ATU-III master enable register (ATUENR) are set to 1, counting is enabled.

TSTRC can be read from and written to in byte or word units.

TSTRC is initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	STRC4	STRC3	STRC2	STRC1	STRC0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	STRC4	0	R/W	Counter C4 Start These bits enable and disable timer counter Cn (TCNTCn) in a subblock. When bit STRCn is cleared to 0, TCNTCn is stopped. While TCNTCn is stopped, the previous counter value is retained and TCNTCn is resumed from the value when this bit is set to 1 again. Note that counting is enabled when both this bit and the TCE bit in ATUENR must be set to 1. 0: TCNTCn is disabled 1: TCNTCn is enabled Note: The prescalers run regardless of this bit and are not synchronized with the timing at which this bit is set. Therefore, the time from when this bit is set to 1 until TCNTCn is incremented for the first time is less than the cycle time of the clock of TCNTCn.
3	STRC3	0	R/W	
2	STRC2	0	R/W	
1	STRC1	0	R/W	
0	STRC0	0	R/W	

Note: n = 0 to 4

13.16.2 Noise Canceler Control Register C0 to C4 (NCCRC0 to NCCRC4)

NCCRC0 to NCCRC4 is an 8-bit readable/writable register. The noise cancellation is performed on the input capture trigger signal input from pin TIOCnm in subblock C0 to C4 by setting this register. Two modes are available in noise cancellation and can be switched by bit NCMCn.

NCCRC0 to NCCRC4 can be read from and written to in byte or word units.

NCCRC0 to NCCRC4 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	NCECn3	NCECn2	NCECn1	NCECn0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
3	NCECn3	0	R/W	Noise Canceler Enable Cn3 to Cn0 These bits enable and disable the noise cancelers for pins TIOCnm in subblock C0 to C4. 0: Noise cancelers for inputs on TIOCnm are disabled 1: Noise cancelers for inputs on TIOCnm are enabled When a level change on externally input signals TIOCnm is detected while this bit is set to 1, it is processed in premature-transition cancellation or minimum time-at-level cancellation mode depending on the setting in the noise cancellation mode register (NCMR) of the common controller. In premature-transition cancellation mode When a level change of the externally input signal is detected, the change is output as the signal whose noise is removed and the corresponding noise canceler counter (NCNTCnm) is started for counting up. Subsequent level changes are masked until the value in the counter reaches the value in the noise canceler register (NCRCnm). The level of the externally input signal is output on this compare match. When these bits are cleared to 0 while the counter (NCNTCnm) is being incremented, counting continues until the values in the counter and the noise canceler register match. The subsequent level changes are also masked over this period. In minimum time-at-level cancellation mode When a level change of the externally input signal is detected, the corresponding noise canceler counter (NCNTCnm) is started for counting up. If subsequent level changes are not detected until the value in the counter reaches the value in the noise canceler register (NCRCnm), the previously accepted level change is output as the signal whose noise is removed on compare match of the counter and noise canceler register. When the subsequent level change is detected before the values in the counter and noise canceler register match, all the changes are treated as noise. Therefore the signal whose noise is removed is not changed. When these bits are cleared to 0 while the counter (NCNTCnm) is being incremented, counting continues until the values in the counter and the noise canceler register match or a level change on the externally input signal is detected. For details on operations in each mode, see figures 13.1 and 13.2.
2	NCECn2	0	R/W	
1	NCECn1	0	R/W	
0	NCECn0	0	R/W	

Note: n = 0 to 4, m = 0 to 3

13.16.3 Timer Control Registers C0 to C4 (TCRC0 to TCRC4)

TCRC0 to TCRC4 are 8-bit readable/writable registers that select the counter clock for subblocks C0 to C4 and enable and disable the PWM mode and the forced compare matches.

TCRC0 to TCRC4 can be read from and written to in byte or word units.

TCRC0 to TCRC4 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	FCMC n3	FCMC n2	FCMC n1	FCMC n0	PWM n0	CKSELCn[2:0]		
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: n = 0 to 4 (correspond to subblocks C0 to C4)

Bit	Bit Name	Initial Value	R/W	Description
7	FCMCn3	0	R/W	Forced Compare Match Cnm Setting these bits to 1 generates forced compare match when a general register (GRCnm) is used for compare match. 0: No forced compare match occurs on general register GRCnm 1: Forced compare match generated on general register GRCnm A compare match flag (IMFCnm) in timer status register Cn (TSRCn) is set to 1 in the cycle following the cycle in which the FCMCnm bit is set to 1. An output level on the TIOCnm signal is also changed in a similar way of normal compare match. While this bit is set to 1, the state of a compare match occurrence is retained regardless of GRCnm and TCNTCn. TCNTCn continues counting up on its counting clock. When the compare match flag bit (IMFCnm) in timer status register Cn (TSRCn) is cleared to 0, this bit is automatically cleared to 0. Any of compare match on a general register and a counter is ignored until this bit is cleared. After this bit is cleared to 0, compare match of TCNTCn and GRCnm can be performed. When both bits PWMn0 and FCMCn0 are set to 1 and forced compare match occurs. At this time, a counter is cleared and a signal is output on pin TIOCnm according to the value in the IOCn0 bits. TCNTCn restarts counting up from H'000000. After that, while bit FCMCn0 is set to 1, another compare match is not generated. (Only a counter of channel 0 in each subblock can be cleared on compare match.)
6	FCMCn2	0	R/W	
5	FCMCn1	0	R/W	
4	FCMCn0	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
3	PWMn0	0	R/W	PWM Mode Setting this bit to 1 makes subblock Cn operate in PWM mode. In PWM mode, TCNTCn is cleared on compare match between TCNTCn and general register GRCn0. The setting of this bit is valid when GRCn0 functions as a compare match register. When general register GRCnm (m = 1 to 3) functions as a compare match register, a signal is output on pin TIOCnm according to the setting in the IOCnm bits in TIORCn. A signal level of a logical one can be output on cycle compare match with GRCn0, and a signal level of a logical zero can be output on duty cycle compare match with GRCn1 to GRCn3. To make the subblock operate in PWM mode, further setting is needed. Select the compare match by the IOCnm bits in timer I/O control register C (TIORCn) for GRCn1 to GRCn3 which operate in PWM mode and GRCn0. Note that a logical zero output on compare match must be selected for GRCn 1 to GRCn3. 0: Subblock Cn does not operate in PWM mode 1: Subblock Cn operates in PWM mode When TCNTCn matches GRCn0 while this bit is set to 1, TCNTCn is cleared. However, when clearing the counter on compare match and incrementation occur simultaneously, TCNTCn is set to H'000001. This occurs when TCNTCn is driven by the clock whose frequency is equal to the Pϕ clock. In PWM mode, do not set GRCn0 to GRCn3 to H'000000. If GRCn0 is set to H'000000, compare match occurs at illegal cycles.

Bit	Bit Name	Initial Value	R/W	Description
2 to 0	CKSELCn [2:0]	000	R/W	TCNTCn Clock Select These bits select the counter clock of subblock Cn. Counters in subblock Cn (TCNTCn) are driven by the clock selected in these bits. 000: Counters are driven by clock-bus line 0 001: Counters are driven by clock-bus line 1 010: Counters are driven by clock-bus line 2 011: Counters are driven by clock-bus line 3 100: Counters are driven by clock-bus line 4 101: Counters are driven by clock-bus line 5 11x: Reserved x denotes don't care.

Note: n = 0 to 4 (n denotes subblock number), m = 0 to 3 (m denotes general register number)

13.16.4 Timer Status Registers C0 to C4 (TSRC0 to TSRC4)

TSRC0 to TSRC4 are 8-bit readable/writable registers indicate occurrence of overflow of timer counter Cn (TCNTCn) in subbblocks C0 to C4 and input capture and compare match of GRC00 to GRC43.

The flags are in this register are used as interrupts. When the corresponding bit in timer interrupt enable register C (TIERC) is set to 1, an interrupt request is issued.

TSRC0 to TSRC4 can be read from and written to in byte or word units.

TSRC0 to TSRC4 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	OVFCn	IMFCn3	IMFCn2	IMFCn1	IMFCn0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: n = 0 to 4 (correspond to subblocks C0 to C4)
* Only 0 can be written to this bit after it is read as 1 to clear it. Writing 1 is ignored.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	OVFCn	0	R/W	Overflow Flag Cn Indicates whether or not TCNTCn has overflowed. This flag cannot be set to 1 by software. 0: TCNTCn has not overflowed [Clearing condition] • When writing 0 to this bit after reading it as 1 1: TCNTCn has overflowed [Setting condition] • When TCNTCn overflows (from H'FFFFFF to H'000000) When TCNTCn is incremented while it is H'FFFFFF, it overflows. When writing H'000000 to TCNTCn or TCNTCn is started from H'000000, this bit is not set to 1. When writing to TCNTCn at the same time as incrementation while it is H'FFFFFF, this bit is set to 1. However, TCNTCn is started from the written value.

Bit	Bit Name	Initial Value	R/W	Description
3 to 0	IMFCn3 to IMFCn0	All 0	R/W	Input Capture/Compare Match Flag Cnm These bits indicate whether or not input capture and compare match between general register GRCnm and TCNTCn has occurred. This flag cannot be set to 1 by software. Setting and clearing conditions are shown below. 0: Neither input capture nor compare match has occurred [Clearing conditions: input capture/output compare] • When writing 0 to this bit after reading it as 1 • When bits in subblocks C00, C10, C20, C30, and C40 are cleared by the ACK signal output from the DMAC [Clearing condition: input capture] • When this bit is automatically cleared on read access to general register GRCnm by the A-DMAC 1: Input capture or compare match has occurred [Setting condition: input capture] • When GRCnm functions as an input capture register and the value in TCNTCn is transferred to GRCnm on an assertion of the input capture signal [Setting conditions: output compare] • When GRCnm functions as a compare match register and the values in TCNTCnm and GRCnm match • When the forced compare match bit (FCMCnm) in TCRCn is set to 1 Even if these bits are set to 1 meaning that the flag has not been cleared, another input capture or output compare signal can be input. A value of 1 is written to these bits. Even if the compare match flag is cleared to 0 while TCNTCn = GRCnm after the compare match is detected, these bits are not set to 1. Bits in subblocks C00, C10, C20, C30, and C40 can be used as the interrupt request to the D-MAC. These bits can be automatically cleared to 0 by the ACK signal.

Notes: n = 0 to 4 (n denotes subblock number), m = 0 to 3 (m denotes channel number)

* Only 0 can be written to this bit after it is read as 1 to clear the flag. Writing 1 to this bit is ignored.

13.16.5 Timer Interrupt Enable Registers C0 to C4 (TIERC0 to TIERC4)

TIERC0 to TIERC4 are 8-bit readable/writable registers that enable and disable interrupt requests for timer C as the input capture, output compare, and overflow-interrupt.

TIERC0 to TIERC4 can be read from and written to in byte or word units.

TIERC0 to TIERC4 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	OVECn	IMECn3	IMECn2	IMECn1	IMECn0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W

Note: n = 0 to 4 (correspond to subblocks C0 to C4)

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	OVECn	0	R/W	Overflow Interrupt Enable Cn Enables and disables an interrupt request when the overflow flag (OVFCn) in timer status register Cn (TSRCn) is set to 1. 0: Disables an OVFCn interrupt request 1: Enables an OVFCn interrupt request
3	IMECn3	0	R/W	Input Capture/Compare Match Interrupt Enable nm These bits enable and disable an interrupt request when the input capture/compare match flag in timer status register Cn (TSRCn). 0: Disables an IMFCnm interrupt request 1: Enables an IMFCnm interrupt request
2	IMECn2	0	R/W	
1	IMECn1	0	R/W	
0	IMECn0	0	R/W	

Note: n = 0 to 4 (n denotes subblock number), m = 0 to 3 (m denotes channel number)

13.16.6 Timer I/O Control Registers C0 to C4 (TIORC0 to TIORC4)

TIORC0 to TIORC4 are 16-bit readable/writable registers.

Compare match and input capture functions are switched by the setting in the IOCnm[2] bit. Before changing the IOCnm[2] bit, clear the IOCnm[1:0] bits to B'00. After that, the output signal level for compare match or the input capture edge for input capture can be set.

When the compare match function is selected, the initial level of the compare match signal is a logical zero. During operation, the previous level is retained. When two functions are switched, the counter must be stopped. Otherwise, operation cannot be guaranteed.

TIORC0 to TIORC4 can be read from and written to in byte- or word-units.

TIORC0 to TIORC4 are initialized to H'0000 by a power-on reset or a transition to the hardware standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	IOCn3[2:0]			-	IOCn2[2:0]			-	IOCn1[2:0]			-	IOCn0[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Note: n = 0 to 4 (correspond to subblocks C0 to C4)

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
14 to 12	IOCn3[2:0]	000	R/W	I/O Control These bits select the function of general register GRCn3. When GRCn3 functions as the output compare register 000: Compare match function disabled 001: Logical zero is output on compare match 010: Logical one is output on compare match 011: Output levels are toggled every compare match When GRCn3 functions as the input capture register 100: Input capture function disabled 101: Input capture on the rising edge of TIOCnm 110: Input capture on the falling edge of TIOCnm 111: Input capture on both edges of TIOCnm
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10 to 8	IOCn2[2:0]	000	R/W	I/O Control These bits select the function of general register GRCn2. When GRCn2 functions as the output compare register 000: Compare match function disabled 001: Logical zero is output on compare match 010: Logical one is output on compare match 011: Output levels are toggled every compare match When GRCn2 functions as the input capture register 100: Input capture function disabled 101: Input capture on the rising edge of TIOCnm 110: Input capture on the falling edge of TIOCnm 111: Input capture on both edges of TIOCnm
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
6 to 4	IOCn1[2:0]	000	R/W	I/O Control These bits select the function of general register GRCn1. When GRCn1 functions as the output compare register 000: Compare match function disabled 001: Logical zero is output on compare match 010: Logical one is output on compare match 011: Output levels are toggled every compare match When GRCn1 functions as the input capture register 100: Input capture function disabled 101: Input capture on the rising edge of TIOCnm 110: Input capture on the falling edge of TIOCnm 111: Input capture on both edges of TIOCnm
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	IOCn0[2:0]	000	R/W	I/O Control These bits select the function of general register GRCn0. When GRCn0 functions as the output compare register 000: Compare match function disabled 001: Logical zero is output on compare match 010: Logical one is output on compare match 011: Output levels are toggled every compare match When GRCn0 functions as the input capture register 100: Input capture function disabled 101: Input capture on the rising edge of TIOCnm 110: Input capture on the falling edge of TIOCnm 111: Input capture on both edges of TIOCnm

Note: n = 0 to 4 (n denotes subblock number), m = 0 to 3 (m denotes channel number)

13.16.7 Timer Counters C0 to C4 (TCNTC0 to TCNTC4)

TCNTC0 to TCNTC4 are 32-bit readable/writable registers driven by the input clock. These counters can be read from and written to while they are being run.

Timer counter Cn (TCNTCn) is started for counting by setting the bit in timer start register C (TSTRC) to 1. The clock signal is selected by the clock select bit (CKSEL) in timer control register Cn (TCRCn). When these counters overflow, the overflow flag (OVFCn) in timer status register Cn (TSRCn) is set to 1.

TCNTC0 to TCNTC4 can be read from and written to in longword units.

TCNTC0 to TCNTC4 are initialized to H'00000000 by a power-on reset or a transition to the hardware standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

13.16.8 General Registers C00 to C43 (GRC00 to GRC43)

GRC00 to GRC43 are 32-bit readable/writable registers that function as the input capture register or output compare register. These functions are switched by setting timer I/O control register Cn (TIORCn).

If these registers function as the input capture register, they capture the value in TCNTCn when the externally input capture signal is detected. At this time, the IMFC bit in timer status register Cn (TSRCn) is set to 1. The edge to be detected is selected by TIORCn.

Input capture is performed even if the counter is stopped (the TCE bit in ATUENR is cleared to 0 or the STRCn bit in TSTRC is cleared to 0). The value in the counter stopped is loaded to GRCnm.

If these registers function as the output compare register, the values in GRCnm and the timer counter (TCNTCn) are constantly compared. The IMFC bit in timer status register (TSRCn) is set to 1 on the first edge of the Pϕ clock after they match. At this time, the level on the TIOCnm pin is changed according to the setting in TIORCn (logical zero, logical one, or toggled). Initially, logical zero is output on the TIOCnm pin (immediately after a reset or output compare modes are switched). During operation, the previous value is output.

GRC00 to GRC43 can be read from and written to in longword units.

GRC00 to GRC43 are initialized to H'FFFF FF00 by a power-on reset or a transition to the hardware standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									-	-	-	-	-	-	-	-
Initial value:	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

13.16.9 Noise Canceler Counters C00 to C43 (NCNTC00 to NCNTC43)

NCNTC00 to NCNTC43 are 8-bit readable/writable registers.

NCNTC00 to NCNTC43 are started by the external input signals TIOC00 to TIOC43 as triggers when the noise canceler is enabled by the noise canceler enable bit (NCEC00 to NCEC43) in timer I/O control register C0 to C4 (TIORC0 to TIORC4). These counters are driven by the Pϕ clock or the Pϕ clock divided by 128 output from the prescaler.

NCNTCnm continues counting regardless of the settings in the TCE bit in ATU-III master enable register (ATUENR) and in TSTRC (regardless of the TCNTCn).

The input signals are processed in premature-transition cancellation or minimum time-at-level cancellation mode depending on the setting of the timer C noise cancellation mode bit in the noise cancellation mode register (NCMR) of the common controller.

- Premature-Transition Cancellation Mode

  When a level change of the externally input signal (TIOCnm) is detected while bit NCECnm is set to 1 and NCNTCnm is stopped, NCNTCnm is started for counting up. This counter is cleared to H'00 and stopped on the first edge of the Pϕ clock after the value in NCNTCnm matches the value in noise canceler register nm (NCRCnm). NCNTCnm is incremented regardless of the TCE bit in ATUENR. The first change is output as the signal whose noise is removed and the edge is to be extracted. Subsequent level changes are masked until the value in the counter reaches the value in the noise canceler register (NCRCnm). The level of the externally input signal is output on this compare match.

  When the NCECnm bit is cleared to 0 while the counter (NCRCnm) is being incremented, counting continues until the values in the counter and the noise canceler register match. The subsequent level changes are also masked over this period.

- In Minimum Time-at-Level Cancellation Mode

  When a level change of the externally input signal (TIOCnm) is detected while the NCECnm bit is set to 1 and NCNTnm is stopped, NCNTnm is started for counting up. This counter is cleared to H'00 and stopped on the first edge of the Pϕ clock after the value in NCNTnm matches the value in noise canceler registers nm (NCRCnm) or after the level of the externally input signal (TIOCnm) is changed. NCNTnm is incremented regardless of the TCE bits in ATUENR.

When a level change of the externally input signal is detected, the corresponding noise canceler counter (NCNTnm) is started for counting up. If subsequent level changes are not detected until the value in the counter reaches the value in the noise canceler register (NCRCnm), the previously accepted level change is output as the signal whose noise is removed on compare match of the counter and noise canceler register. When the subsequent level change is detected before the values in the counter and noise canceler register match, all the changes is treated as noise. Therefore the signal whose noise is removed is not changed.

When the NCECnm bits are cleared to 0 while the counter (NCNTCnm) is being incremented, counting continues until the values in the counter and the noise canceler register match or a level change on the externally input signal is detected.

NCNTCnm can be written to regardless of the operating state and can start from the value rewritten.

NCNTC00 to NCNTC43 can be read from and written to in byte or word units.

NCNTC00 to NCNTC43 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	NCNTCnm[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Notes: 1. n = 0 to 4
2. m = 0 to 3 (correspond to GRC00 to GRC43)

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	NCNTCnm[7:0]	All 0	R/W	Noise Cancel Count Cnm These bits store an 8-bit count value.

Note: n = 0 to 4. m = 0 to 3: GRC00 to GRC43

13.16.10 Noise Cancel Registers C00 to C43 (NCRC00 to NCRC43)

NCRC00 to NCRC43 are 8-bit readable/writable registers that are provided in each subblock and set the upper limitations of noise canceler counters C00 to C43 (NCNTC00 to NCNTC43).

A pulse width of up to 1.64 ms (= 50 ns × 128 division × 256 counts) can be treated as noise by setting the registers to H'FF.

Input edges are processed in premature-transition cancellation or minimum time-at-level cancellation mode depending on the setting of the NCMC bit in the noise cancellation mode register (NCMR) of the common controller.

- Premature-Transition Cancellation Mode

  While NCNTCnm is in count operation, the level change of the subsequent input signal is masked. Values in NCNTCnm and NCRCnm are always compared. If a compare match occurs, the value in NCNTCnm is cleared on the next Pϕ clock, the count operation is stopped, and the masking of the input signal is canceled.
- In Minimum Time-at-Level Cancellation Mode

  While NCNTCnm is in count operation, noise canceler processing waiting state is entered. Values in NCNTCnm and NCRCnm are always compared. If a compare match occurs, the value in NCNTCnm is cleared on the next Pϕ clock, the count operation is stopped, and at the same time the noise canceler outputs the input signal that has passed through the noise canceling processing.

NCRC00 to NCRC43 can be read from and written to in byte or word units.

NCRC00 to NCRC43 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	NCRCnm[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Notes: 1. n = 0 to 4
2. m = 0 to 3 (correspond to GRC00 to GRC43)

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	NCRCnm[7:0]	All 0	R/W	Noise Cancellation Time Cnm TIOCnm noise cancel period (8-bit compare value)

13.17 Operations of Timer C

13.17.1 Input Capture Function

General registers GRCnm of timer C perform input capture operation when the input capture operation is selected by timer I/O control register Cn (TIORCn). Capturing is perfumed when an edge on external input pins (TIOCnm) is detected.

Timer counter Cn (TCNTCn) is started for counting up by setting a bit in timer start register C (TSTRC) to 1. When an edge on an external pin corresponding to GRCnm is detected, a bit (IMFCnm) in timer status register C (TSRC) is set to 1 and the counter value is transferred to GRCnm. This flag indicating that an input capture has occurred and an interrupt request signal are changed two cycles of the Pϕ clock after the edge on pin TIOCnm is detected.

The edge type is selected from the rising, falling, or both by the IOCnm[2:0] bits in TIORCn. An interrupt request can be output by setting timer interrupt enable register Cn (TIERCn).

The input capture flag (the IMFCnm bits in TSRCn) is cleared; when 0 is written to the bits after they are read as 1; when GRCnm is read by the A-DMAC (automatic clearing); when an ACK signal is input by the DMAC.

Figure 13.24 shows an operation example of input capture in subblock C0. In this example, the both edges on pin TIOC00, the rising edge on pin TIOC01, and the falling edge on TIOC02 are selected.

Input capture is performed even if TCNTCn is stopped (when the TCE bit in ATUENR or bit STRCn is cleared to 0). The counter value of the TCNTCn stopped is captured in GRCnm.

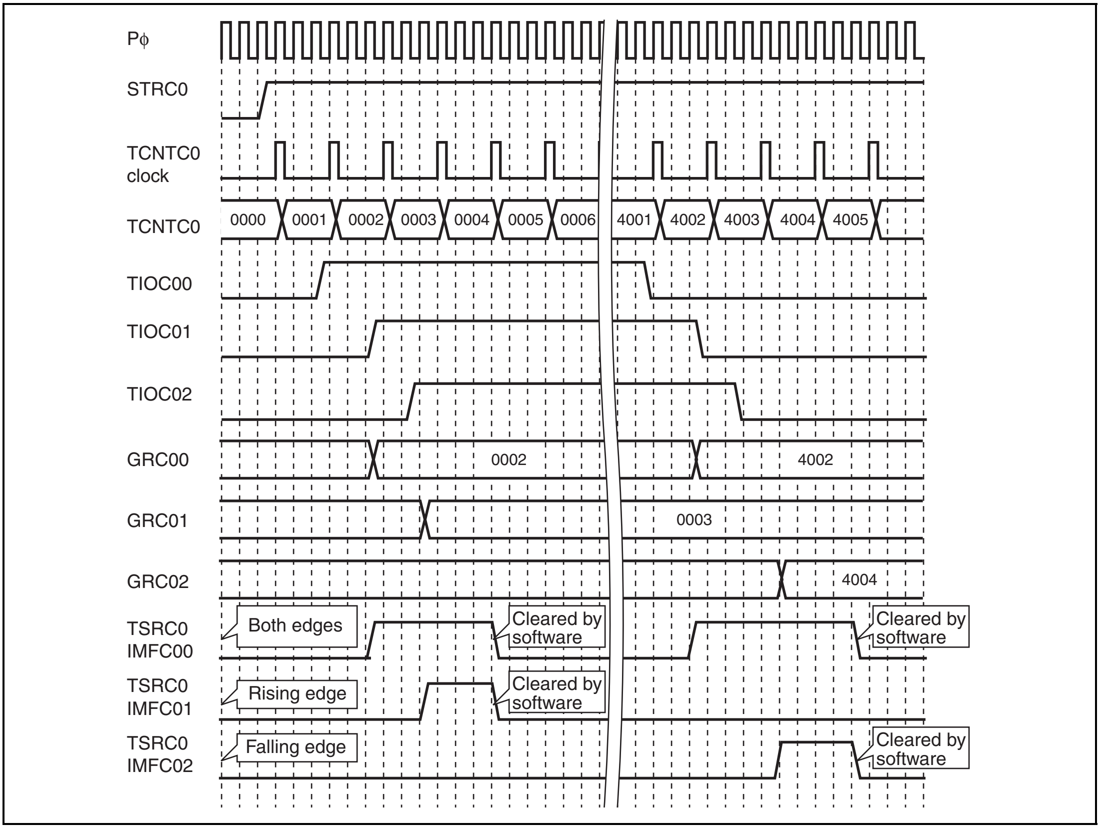


Figure 13.24 Operation Example of Input Capture in Subblock C0

13.17.2 Compare Match Function

A compare match signal can be output on external pin TIOCnm by setting general register GRCnm as the compare match register in timer I/O control register Cn (TIORCn).

Timer counter Cn (TCNTCn) is started for counting up by setting a bit in timer start register C (TSTRC) to 1. Set the value in GRCnm before starting the counter. When the values in GRCnm and TCNTCn match, a bit corresponding to GRCnm in timer status register Cn (TSRCn) is set and a waveform is output on external pin TIOCnm.

The compare match flag is set and the signal level on pin TIOCnm is changed on the first edge of the Pϕ clock immediately after compare match between GRCnm and TCNTCn.

A logical one, a logical zero, or a toggled output can be selected for the signal to be output on pin TIOCnm.

An interrupt can be output by setting timer interrupt enable register C (TIERC).

Figure 13.25 shows an operation example of compare match. In this example, a toggled output on GRC00, a logical one output on GRC01, and a logical zero output on GRC02 are externally output. H'004004 is set in GRC0m. The status flag and the output level on pin TIOC0m are changed on the first edge of the Pϕ clock immediately after compare match between GRCnm and TCNTCn.

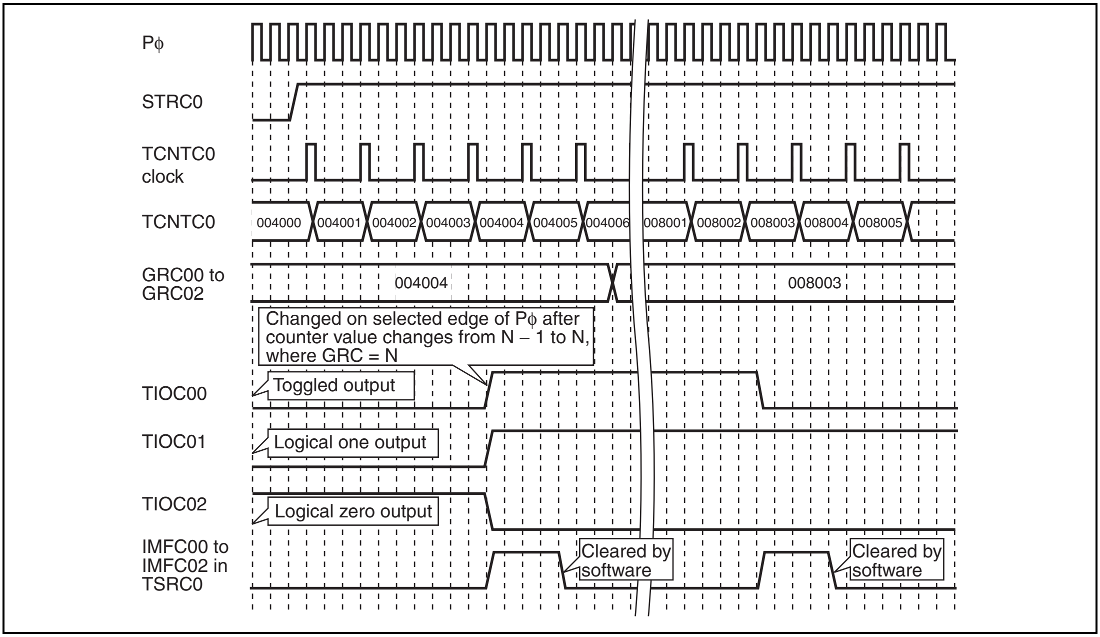


Figure 13.25 Operation Example of Compare Match

By setting a forced compare match bit (FMCMnm) in timer control register Cn (TCRCn), compare match can be generated even if TCNTCn has not matched GRCnm. The compare match flag and the signal level on pin TIOCnm are changed on the first edge of the Pϕ clock after bit FCMCnm is set to 1. Compare match of TCNTCn with GRCnm after a forced compare match is not performed until the compare match flag (IMFCnm) is cleared. Bit FCMCnm is cleared at the same time as bit IMFCnm in TSRCn is cleared.

To clear the compare match flag, write 0 to the status flag after it is read as 1. GRC00, GRC10, GRC20, GRC30, and GRC40 can be cleared by an interrupt request from the DMAC.

Compare match is detected when any of the following occurs.

- When the values in TCNTCn and GRCnm match (other than after forced compare match)

- When the forced compare match bit (bit FCMCnm in TCRCn) is changed from 0 to 1
- When the values in TCNTCn and GRCnm match because a counter is cleared on compare match of GRCn0

The compare match flag and the signal level on pin TIOCnm is changed on the first edge of the Pϕ clock after bit FCMCnm is set to 1. Make sure that the compare match operation is selected by bits IOCnm[2:0] in TIORCn before starting operation. Compare match is not detected; when GRCnm is set to the same value as TCNTCn; when the compare match operation is selected after setting the forced compare match bit to 1.

Compare match is detected regardless of the counter operating state. Even if the counter is stopped, compare match occurs when the condition is satisfied.

When the compare match status flag is cleared before GRCnm and TCNTCn are changed (such as before counting while the counter has been stopped), the compare match is not detected.

13.17.3 PWM Function

Setting bit PWMn in timer control register Cn (TCRCn) to 1 makes channels 1 to 3 in each subblock function as PWM timers with the same frequency. In PWM mode, GRCn0 as a cycle setting register and GRCn1 to GRCn3 as duty cycle setting registers are used. External pins TIOCn1 to TIOCn3 corresponding GRCn1 to GRCn3 are used to output PWM signals. To use them as PWM signal outputs, select the compare match operation by bits IOCnm in TIORCn as well as setting bit PWMn so that GRCn0 to GRCn3 function as compare match registers.

Timer counter Cn (TCNTCn) is started by setting timer start register C (TSTRC). When the value in TCNTCn reaches the value in the cycle setting register (GRCn0), compare match occurs and the bit in timer status register C (TSRCn) is set to 1. At this time, TCNTCn is cleared and a signal is output on external pins TIOCn1 to TIOCn3 according to bit IOCn0 in PWM mode. The output signal levels on pin TIOCn0 depend on bit IOCn0.

When the value in TCNTCn reaches the value in the duty cycle setting register (GRCn1 to GRCn3), the bit in timer status register Cn (TSRCn) is set to 1 and a signal is output on external pins TIOCn1 to TIOCn3 according to bits IOCn1 to IOCn3.

When the same value is set in the cycle and duty cycle setting registers, priority is given to bit IOCn0 corresponding to the cycle setting register.

Figure 13.26 shows an operation example of subblock C0 in PWM mode.

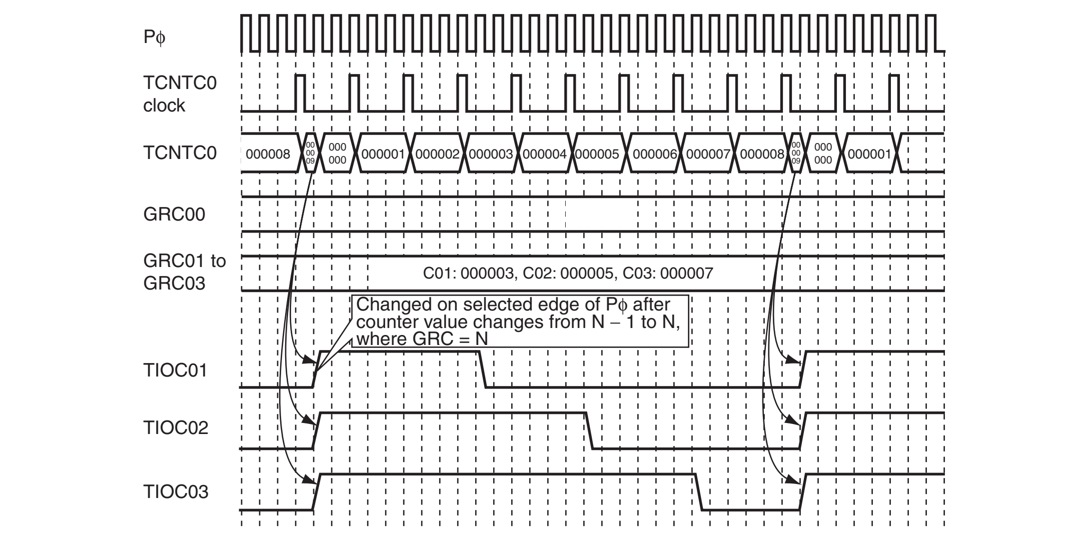


Figure 13.26 Operation Example of Subblock C0 in PWM Mode

13.18 Overview of Timer D

Timer D consists of four subblocks that output one-shot pulses. The subblocks are identical each other.

Timer D has the following functions.

- Down counter is started on compare match A of the output compare register, compare match B of the general register, or writing to the counter start bit in the down-counter start register. A one-shot pulse with an offset can be output.
- Output of a waveform can be forcibly terminated regardless of the down-counter value (priority is given to output termination over compare match A or writing to the counting start bit)
- General register can capture the value in TCNT2Dn on compare match A as a trigger
- Pulse indicating that compare match A or B has been detected for A/D converter activation can be output (16 lines supported by subblocks D0 and D1)
- Interrupt requests can be output on compare matches A and B (16 lines supported). The output signal is ORed on compare matches A and B.
- Interrupt requests can be output on counter overflow. Four outputs from TCNT1Dn and four outputs from TCNT2Dn.
- Interrupt requests can be output on down-counter underflow (16 lines supported). A DMA transfer request can be issued for the DMAC. In this case, the compare match flag can be cleared by the ACK signal from the DMAC. (16 lines for DMA transfer requests supported by down counters in subblocks D0 to D3)
- Offset base register can capture the counter value by a trigger from timer A

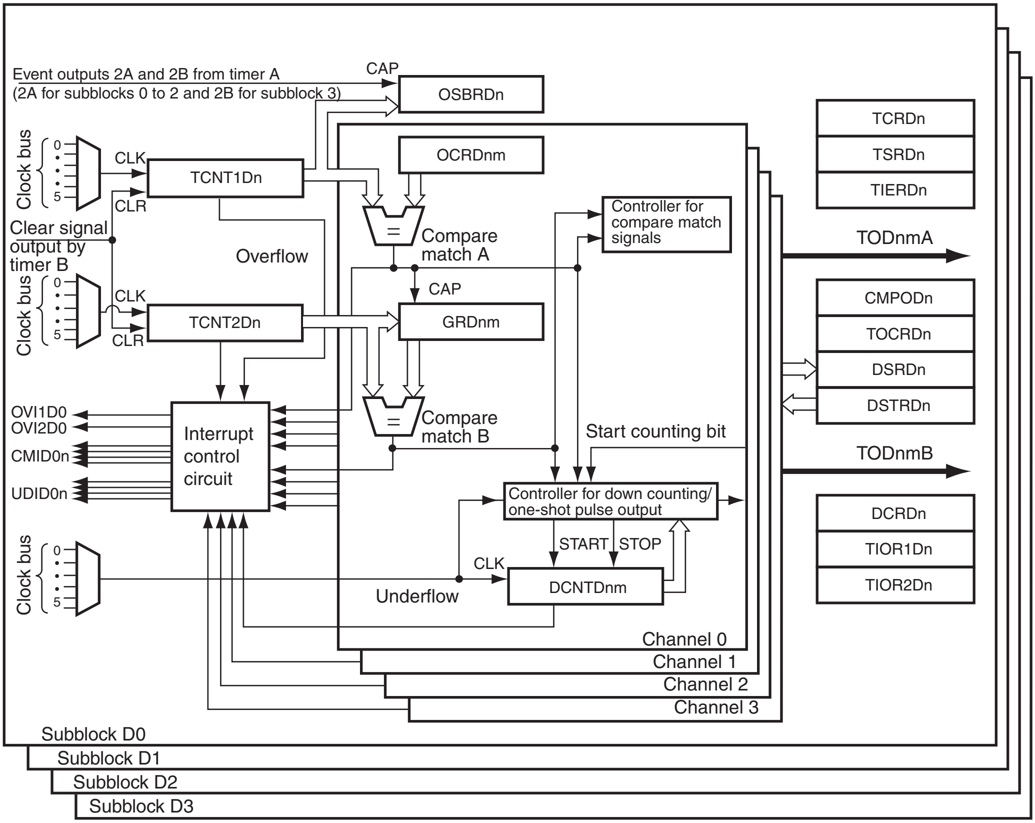
13.18.1 Block Diagram of Timer D

Each subblock consists of two timer counters (TCNT1Dn and TCNT2Dn), one offset base register (OSBRDn), four output compare registers (OCRDnm), four general registers (GRDnm), four timer down counters (DCNTDnm), and controller. Each channel includes two output pins; TODnmA for compare match and TODnmB for one-shot pulse output.

TODnmA and TODnmB are output a level of 0 as a default.

A trigger for activating the A/D converter can be output to the on compare matches A and B.

Figure 13.27 is a block diagram of timer D.



Note: n = 0 to 3 (n denotes subblock number), m = 0 to 3 (m denotes channel number)

[Legend]

TCNT:	Timer counter	CMPOD:	Compare match pulse output control register D
OSBRD:	Timer offset base register D	TOCRD:	Timer output control register D
OCRD:	Output compare register D	DSRD:	Down counter status register D
GRD:	General register D	DSTRD:	Down counter starting register D
TCRD:	Timer control register D	DCRD:	Down counter control register D
TSRD:	Timer status register D	TIOR:	Timer I/O control register
TIERD:	Timer interrupt enable register D		

Figure 13.27 Block Diagram of Timer D

13.19 Description of Timer D Registers

13.19.1 Timer Start Register (TSTRD)

TSTRD is an 8-bit readable/writable register that enables and disables two timer counters (TCNT1Dn and TCNT2Dn) and timer down counters (DCNTDnm) in subblocks D0 to D3. When the counter Dn start bit and the TDE bit in the ATU-III master enable register (ATUENR) are both set to 1, the counters are started.

TSTRD can be read from and written to in byte or word units.

TSTRD is initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	STRD3	STRD2	STRD1	STRD0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	STRD3	0	R/W	Counter Dn Start These bits enable and disable timer counters 1Dn and 2Dn (TCNT1Dn, TCNT2Dn) and timer down counters (DCNTDnm). The counter value is retained while the counter is stopped. When this bit is set to 1 again, the counter is restarted from the value. Note that this bit and the TDE bit in ATUENR are both set to 1 to restart the counter. 0: TCNT1Dn, TCNT2Dn, and DCNTDnm are disabled 1: TCNT1Dn, TCNT2Dn, and DCNTDnm are enabled The prescalers run regardless of this counter Dn start bit and are not synchronized with the timing at which this bit is set. Therefore, the time from when this bit is set to when TCNT1Dn and TCNT2Dn are incremented for the first time is less than the cycle of the clock of TCNT1Dn and TCNT2Dn.
2	STRD2	0	R/W	
1	STRD1	0	R/W	
0	STRD0	0	R/W	

Note: n = 0 to 3 (n denotes subblock number), m = 0 to 3 (m denotes channel number)

13.19.2 Timer Control Registers D0 to D3 (TCRD0 to TCRD3)

TCRD0 to TCRD3 are 16-bit readable/writable registers that select the counter clocks in subblock Dn for timer counter 1 (TCNT1Dn), timer counter 2 (TCNT2Dn), and timer down counter (DCNTDnm) from clock-bus lines 0 to 5. These registers also enable and disable capture of the timer offset base register and counter clearing requests output from timer B for TCNT1Dn and TCNT2Dn.

TCRD0 to TCRD3 can be read from and written to in byte- or word-units.

TCRD0 to TCRD3 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	OBR EDn	C2C EDn	C1C EDn	-	CKSEL2Dn[2:0]			-	CKSEL1Dn[2:0]			-	DCSELDn[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Note: n = 0 to 3 (correspond to subblocks D0 to D3)

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14	OBREDn	0	R/W	Timer Offset Base Register Enable Enables and disables capture of timer offset base register Dn (OSBRDn). When this bit is set to 1, the value in TCNT1Dn is captured by OSBRDn in the Pϕ clock cycle following cycle in which an event from timer A is asserted. When the pulse width of the event exceeds one cycle of the Pϕ clock, the counter value is captured every clock cycle. 0: Input capture by OSBRDn is enabled 1: Input capture by OSBRDn is disabled

Bit	Bit Name	Initial Value	R/W	Description
13	C2CEDn	0	R/W	Counter 2 Clear Enable Enables and disables clearing the value in TCNT2Dn by timer B. When an edge of the counter clearing signal output from timer B is detected while this bit is set to 1, TCNT2Dn is cleared in the following timing. When the rising edge of the clearing signal is detected in the cycle in which TCNT2Dn is counted up, the counter is cleared on the counting up timing. When the rising edge of the clearing signal is detected in other than the TCNT2Dn counting-up cycle, the counter is cleared on the first counting up timing after edge detection. The counter clearing signal is ignored with this bit set to the initial value. 0: TCNT2Dn clearing signal from timer B is disabled 1: TCNT2Dn clearing signal from timer B is enabled
12	C1CEDn	0	R/W	Counter 1 Clear Enable Enables and disables clearing the value in TCNT1Dn by timer B. When an edge of the counter clearing signal output from timer B is detected while this bit is set to 1, TCNT1Dn is cleared in the following timing. When the rising edge of the clearing signal is detected in the cycle in which TCNT1Dn is counted up, the counter is cleared on the counting up timing. When the rising edge of the clearing signal is detected in other than the TCNT1Dn counting-up cycle, the counter is cleared on the first counting up timing after edge detection. The counter clearing signal is ignored with this bit set to the initial value. 0: TCNT1Dn clearing signal from timer B is disabled 1: TCNT1Dn clearing signal from timer B is enabled
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
10 to 8	CKSEL2Dn [2:0]	000	R/W	TCNT2Dn Clock Select These bits select the TCNT2Dn counting-up clock. 000: Incrementation of TCNT2Dn is driven by clock-bus line 0. 001: Incrementation of TCNT2Dn is driven by clock-bus line 1. 010: Incrementation of TCNT2Dn is driven by clock-bus line 2. 011: Incrementation of TCNT2Dn is driven by clock-bus line 3. 100: Incrementation of TCNT2Dn is driven by clock-bus line 4. 101: Incrementation of TCNT2Dn is driven by clock-bus line 5. 11x: Reserved
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6 to 4	CKSEL1Dn [2:0]	000	R/W	TCNT1Dn Clock Select These bits select the TCNT1Dn counting-up clock. 000: Incrementation of TCNT1Dn is driven by clock-bus line 0. 001: Incrementation of TCNT1Dn is driven by clock-bus line 1. 010: Incrementation of TCNT1Dn is driven by clock-bus line 2. 011: Incrementation of TCNT1Dn is driven by clock-bus line 3. 100: Incrementation of TCNT1Dn is driven by clock-bus line 4. 101: Incrementation of TCNT1Dn is driven by clock-bus line 5. 11x: Reserved

Bit	Bit Name	Initial Value	R/W	Description
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	DCSELDn [2:0]	000	R/W	DCNTDnm Clock Select These bits select the DCNTDnm counting-down clock. Output of one-shot pulse (TODnmB) is synchronized with the clock selected in these bits. 000: Decrementation of DCNTDnm is driven by clock-bus line 0. 001: Decrementation of DCNTDnm is driven by clock-bus line 1. 010: Decrementation of DCNTDnm is driven by clock-bus line 2. 011: Decrementation of DCNTDnm is driven by clock-bus line 3. 100: Decrementation of DCNTDnm is driven by clock-bus line 4. 101: Decrementation of DCNTDnm is driven by clock-bus line 5. 11x: Reserved

[Legend]

x: Don't care

Note: n = 0 to 3 (n denotes subblock number), m = 0 to 3 (m denotes channel number). Counters in channels 0 to 3 of the same subblock use the same clock signal.

13.19.3 Timer I/O Control Registers 1D0 to 1D3 (TIOR1D0 to TIOR1D3)

TIOR1D0 to TIOR1D3 are 16-bit readable/writable registers that select the source of compare match output (TODnmA), enable and disable compare match of OCRDnm, and set the output level on pin TODnA on compare match A.

TIOR1D0 to TIOR1D3 can be read from and written to in byte- or word-units.

TIOR1D0 to TIOR1D3 are initialized to H'0000 by a power-on reset or a transition to the hardware standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OSSDn3[1:0]		OSSDn2[1:0]		OSSDn1[1:0]		OSSDn0[1:0]		IOADn3[1:0]		IOADn2[1:0]		IOADn1[1:0]		IOADn0[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: n = 0 to 3 (correspond to subblocks D0 to D3)

Bit	Bit Name	Initial Value	R/W	Description
15, 14	OSSDn3[1:0]	00	R/W	Compare Match Output Source Select These bits select the output level on the compare match output pin (TODnmA). The output is controlled by compare match A, compare match B, or both matches. Example 1 Compare match A or B is used as a trigger of timer down counter D (DCNTDnm). The other compare match is used as a trigger of compare match output. Example 2 Compare matches A or B is used as a trigger of an assertion of the output. The other compare match is used as a trigger of an negation of the output. A one-shot pulse can be output on pin TODnmA. When both matches are used as a trigger of the output and they occur at the same time, priority is given to compare match B and the output level depends on the IOBnm bit in TIOR2Dn. A level of 0 is output on TODnmA as a default. While these bits are set to B'00, the output level on TODnmA is not changed even if compare match A or B occurs. 00: Output level on TODnmA is not changed 01: Output level on TODnmA depends on the I/O control bit A on compare match A 10: Output level on TODnmA depends on the I/O control bit B on compare match B 11: Output level on TODnmA depends on the I/O control bit A or B on compare match A or B, respectively
13, 12	OSSDn2[1:0]	00	R/W	
11, 10	OSSDn1[1:0]	00	R/W	
9, 8	OSSDn0[1:0]	00	R/W	

Bit	Bit Name	Initial Value	R/W	Description
7, 6	IOADn3[1:0]	00	R/W	I/O Control A These bits select the function of the output compare register (OCRDnm). When these bits are set to B'00, compare match between OCRDnm and timer counter 1 (TCNT1Dn) is not performed. Otherwise, the compare match is performed. When the CMEADnm bit in timer interrupt enable register (TIERDn) is set to 1, an interrupt request is issued on compare match. When the CMPADnm bit in the compare match pulse output control register is set to 1, a trigger to activate the A/D converter is output (supported only by subblocks D0 and D1). If compare match A is selected by the compare match output source select bit (OSSDnm), a signal is output on pin TODnmA according to the IOADnm bits. 00: Compare match is not performed 01: Output level on compare match is 0 10: Output level on compare match is 1 11: Output level on compare match is toggled
5, 4	IOADn2[1:0]	00	R/W	
3, 2	IOADn1[1:0]	00	R/W	
1, 0	IOADn0[1:0]	00	R/W	

Note: n = 0 to 3 (n denotes subblock number), m = 0 to 3 (m denotes channel number)

13.19.4 Timer I/O Control Registers 2D0 to 2D3 (TIOR2D0 to TIOR2D3)

TIOR2D0 to TIOR2D3 are 16-bit readable/writable registers that select the function of general registers (GRDnm). GRDnm can function as capture or compare match registers. TIOR2Dn also enables and disables compare match and set the output level on pin TODnA on compare match.

The GRDnm function can be selected from the capture or compare match by bit I/O control B (IOBDnm).

When TIOR2Dn is used as the compare match register (IOBDnm[2] = 0) and the IOBDnm[1:0] bits are set to B'00, compare match between GRDnm and timer counter 2 (TCNT2Dn) is not performed. Otherwise, the compare match is performed. When the CMEBDnm bit in timer interrupt enable register (TIERDn) is set to 1, an interrupt request is issued on compare match. When the CMPBDnm bit in the compare match pulse output control register is set to 1, a trigger to activate the A/D converter is output (supported only by subblocks D0 and D1).

When compare match B is selected as the output source in the compare match output source select bit (OSSDnm), a signal is output on pin TODnmA according to the IOBDnm bits.

When TIOR2Dn is used as the capture register (IOBDnm[2:0] = B'101), it captures the value in timer counter 2 (TCNT2Dn) on compare match A. Even if TCNT2Dn is stopped, TIOR2D captures the value in TCNT2Dn on compare match A. However, when IOBDnm[2:0] = B'100, TIOR2D does not capture.

When the function is changed while TCNT2Dn = GRDnm, compare match B does not occur. For example, when a general register captures the value in TCNT2Dn, compare match B does not occur even if the function is changed.

When a general register is used as the compare match register and compare match B is selected by the OSSDnm bit, output level on pin TODnA is not changed even if the function is changed to capture, and vice versa.

TIOR2D0 to TIOR2D3 can be read from and written to in byte- or word-units.

TIOR2D0 to TIOR2D3 are initialized to H'0000 by a power-on reset or a transition to the hardware standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	IOBDn3[2:0]			-	IOBDn2[2:0]			-	IOBDn1[2:0]			-	IOBDn0[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Note: n = 0 to 3 (correspond to subblocks D0 to D3)

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 12	IOBDn3 [2:0]	000	R/W	I/O Control B When GRDn3 functions as the output compare register 000: Compare match is not performed 001: Output level on compare match is 0 010: Output level on compare match is 1 011: Output level on compare match is toggled When GRDn3 functions as the capture register 100: Input capture is not performed 101: TCNT2Dn is captured on compare match A 110: Setting prohibited 111: Setting prohibited
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10 to 8	IOBDn2 [2:0]	000	R/W	I/O Control B When GRDn2 functions as the output compare register 000: Compare match is not performed 001: Output level on compare match is 0 010: Output level on compare match is 1 011: Output level on compare match is toggled When GRDn2 functions as the capture register 100: Input capture is not performed 101: TCNT2Dn is captured on compare match A 110: Setting prohibited 111: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6 to 4	IOBDn1 [2:0]	000	R/W	I/O Control B When GRDn1 functions as the output compare register 000: Compare match is not performed 001: Output level on compare match is 0 010: Output level on compare match is 1 011: Output level on compare match is toggled When GRDn1 functions as the capture register 100: Input capture is not performed 101: TCNT2Dn is captured on compare match A 110: Setting prohibited 111: Setting prohibited
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	IOBDn0 [2:0]	000	R/W	I/O Control B When GRDn0 functions as the output compare register 000: Compare match is not performed 001: Output level on compare match is 0 010: Output level on compare match is 1 011: Output level on compare match is toggled When GRDn0 functions as the capture register 100: Input capture is not performed 101: TCNT2Dn is captured on compare match A 110: Setting prohibited 111: Setting prohibited

Note: n = 0 to 3 (n denotes subblock number), m = 0 to 3 (m denotes channel number)

13.19.5 Down Counter Starting Registers D0 to D3 (DSTRD0 to DSTRD3)

DSTRD0 to DSTRD3 are 8-bit readable/writable registers that start the down counter. Setting the bits makes the down counter start.

DSTRD0 to DSTRD3 can be read from and written to in byte or word units.

DSTRD0 to DSTRD3 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	DSTD n3	DSTD n2	DSTD n1	DSTD n0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: n = 0 to 3 (correspond to subblocks D0 to D3)
* Writing 0 is ignored. This bit is always read as 0 even if 1 is written.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	DSTDn3	0	R/(W)*	Down Counter Start Dnm Setting these bits to 1 makes down counter Dn (DCNTDnm) start. The setting in these bits is always valid and regardless of the start trigger setting in the down counter control register. When compare match B and writing 1 to these bits occurs at the same time if the down counter is set so that it is stopped on compare match B, compare match B takes priority and the down counter is not started. When DCNTDn = H'000000, writing 1 to these bits has no effect. 0: No operation 1: Down counters (DCNTDn0 to DCNTDn3) are started
2	DSTDn2	0	R/(W)*	
1	DSTDn1	0	R/(W)*	
0	DSTDn0	0	R/(W)*	

13.19.6 Down Counter Status Registers D0 to D3 (DSRD0 to DSRD3)

DSRD0 to DSRD3 is an 8-bit readable/writable register that indicates the state of the timer down counter (DCNTDnm).

DSRD0 to DSRD3 can be read from and written to in byte or word units.

DSRD0 to DSRD3 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	DSFD n3	DSFD n2	DSFD n1	DSFD n0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R*	R*	R*	R*

Note: n = 0 to 3 (correspond to subblocks D0 to D3)
* DSDFDnm is a read-only bit and cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
3	DSFDn3	0	R*	Down Counter Status Flag Dnm These bits indicate enabling/disabling of down counter Dnm (DCNTDnm). When these bits are read as 1, the counter operation is enabled. If the TDE bit of ATUENR is 1 and the STRDn bit of TSTRD is 1, the counter is running. When these bits are read as 0, the counter operation is disabled, so counting cannot be in progress. 0: Down counter is disabled [Clearing conditions] • When the down counter is stopped by underflow • When the condition to stop the down counter (compare match B) 1: Down counter is enabled [Setting conditions] • When writing 1 to the down counter start bit in the down counter starting register (DSTRDn) • When the condition set as the down counter start trigger is satisfied (compare match A or B) These flags are set or cleared regardless of the settings of the TDE bit in ATUENR and the STRDn bit in TSTRD. Accordingly, if the TDE bit and the STRDn bit are not set to enable counting, the down counter is not actually running even if these bits indicate that counting is enabled.
2	DSFDn2	0	R*	
1	DSFDn1	0	R*	
0	DSFDn0	0	R*	

13.19.7 Down Counter Control Registers D0 to D3 (DCRD0 to DCRD3)

DCRD0 to DCRD3 are 16-bit readable/writable registers that starts the timer down counter (DCNTDnm). Starting and stopping by compare match A or B can be set. To change the TRGSELDn bit, stop the counter. Otherwise operation cannot be guaranteed.

The trigger source to start and stop the counter can be selected by the TRGSELD bits. The counter can be started by compare match A or B and can be stopped by compare match B.

The counter can also be started by writing 1 to the down counter start bit. The trigger source is always valid regardless of the TRGSELD bits and takes priority over other sources.

DCRD0 to DCRD3 can be read from and written to in byte- or word-units.

DCRD0 to DCRD3 are initialized to H'0000 by a power-on reset or a transition to the hardware standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	TRGSELDn3[2:0]			-	TRGSELDn2[2:0]			-	TRGSELDn1[2:0]			-	TRGSELDn0[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Note: n = 0 to 3 (correspond to subblocks D0 to D3)

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 12	TRGSELDn3 [2:0]	000	R/W	Down Counter Start/Stop Trigger Select Dn3 [Counter start trigger] [Counter stop trigger] 000: No trigger 000: No trigger 001: No trigger 001: Compare match B 010: Compare match A 010: No trigger 011: Compare match A 011: Compare match B 100: Compare match B 100: No trigger 101: Setting prohibited 101: Setting prohibited 110: Setting prohibited 110: Setting prohibited 111: Setting prohibited 111: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10 to 8	TRGSELDn2 [2:0]	000	R/W	Down Counter Start/Stop Trigger Select Dn2 [Counter start trigger] [Counter stop trigger] 000: No trigger 000: No trigger 001: No trigger 001: Compare match B 010: Compare match A 010: No trigger 011: Compare match A 011: Compare match B 100: Compare match B 100: No trigger 101: Setting prohibited 101: Setting prohibited 110: Setting prohibited 110: Setting prohibited 111: Setting prohibited 111: Setting prohibited
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6 to 4	TRGSELDn1 [2:0]	000	R/W	Down Counter Start/Stop Trigger Select Dn1 [Counter start trigger] [Counter stop trigger] 000: No trigger 000: No trigger 001: No trigger 001: Compare match B 010: Compare match A 010: No trigger 011: Compare match A 011: Compare match B 100: Compare match B 100: No trigger 101: Setting prohibited 101: Setting prohibited 110: Setting prohibited 110: Setting prohibited 111: Setting prohibited 111: Setting prohibited
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description	
2 to 0	TRGSELDn0 [2:0]	000	R/W	Down Counter Start/Stop Trigger Select Dn0	
				[Counter start trigger]	[Counter stop trigger]
				000: No trigger	000: No trigger
				001: No trigger	001: Compare match B
				010: Compare match A	010: No trigger
				011: Compare match A	011: Compare match B
				100: Compare match B	100: No trigger
				101: Setting prohibited	101: Setting prohibited
				110: Setting prohibited	110: Setting prohibited
				111: Setting prohibited	111: Setting prohibited

Note: n = 0 to 3 (n denotes subblock number)

13.19.8 Timer Status Registers D0 to D3 (TSRD0 to TSRD3)

TSRD0 to TSRD3 are 16-bit readable/writable registers that control overflow on timer counters (TCNT2Dn and TCNT1Dn) in subblocks D0 to D3, underflow on timer down counters (DCNTDnm) in channels 0 to 3, and compare match on output compare register (OCRDnm) and general register (GRDnm)

Interrupt requests can be issued by using overflow flags 2 and 1, underflow flag, and compare match flags A and B. To issue interrupt requests, set the corresponding bit in timer interrupt enable register (TIERDn). By using compare matches A and B, pulses can be output to the A/D converter to start conversion. Eight pulse output lines are available in subblocks D0 and D1.

TSRD0 to TSRD3 can be read from and written to in byte- or word-units.

TSRD0 to TSRD3 are initialized to H'0000 by a power-on reset or a transition to the hardware standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	OVF 2Dn	OVF 1Dn	UDF Dn3	UDF Dn2	UDF Dn1	UDF Dn0	CMF ADn3	CMF ADn2	CMF ADn1	CMF ADn0	CMF BDn3	CMF BDn2	CMF BDn1	CMF BDn0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: n = 0 to 3 (correspond to subblocks D0 to D3)

* Only 0 can be written to this bit after it is read as 1 to clear it. Writing 1 is ignored.

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	OVF2Dn	0	R/(W)*	Overflow Flag 2Dn Indicates whether or not timer counter 2Dn (TCNT2Dn) has overflowed. This flag cannot be set to 1 by software. This bit is set to 1 when TCNT2Dn is incremented while it is H'FFFFFF. Writing H'000000 to TCNT2Dn or starting TCNT2Dn from a initial value of H'000000 has no effect on this bit. When writing to TCNT2Dn at the same time as incrementation while it is H'FFFFFF, this bit is set to 1. However, TCNT2Dn is started from the written value. When an assertion of counter clearing signal from timer B and overflow occur, overflow does not occur. The overflow can be notified as an overflow interrupt request by setting the overflow enable flag (OVE2Dn) in timer interrupt enable register Dn (TIERDn). 0: TCNT2Dn has not overflowed [Clearing condition] • When writing 0 to this bit after reading it as 1 1: TCNT2Dn has overflowed [Setting condition] • When TCNT2Dn overflowed (from H'FFFF FF to H'0000 00)

Bit	Bit Name	Initial Value	R/W	Description
12	OVF1Dn	0	R/(W)*	Overflow Flag 1Dn Indicates whether or not timer counter 1Dn (TCNT1Dn) has overflowed. This flag cannot be set to 1 by software. This bit is set to 1 when TCNT1Dn is incremented while it is H'FFFFFF. Writing H'000000 to TCNT1Dn or starting TCNT1Dn from a initial value of H'000000 has no effect on this bit. When writing to TCNT1Dn at the same time as incrementation while it is H'FFFFFF, this bit is set to 1. However, TCNT1Dn is started from the written value. When an assertion of counter clearing signal from timer B and overflow occur, overflow does not occur. The overflow can be notified as an overflow interrupt request by setting the overflow enable flag (OVE1Dn) in timer interrupt enable register Dn (TIERDn). 0: TCNT1Dn has not overflowed [Clearing condition] • When writing 0 to this bit after reading it as 1 1: TCNT1Dn has overflowed [Setting condition] • When TCNT1Dn overflowed (from H'FFFF FF to H'0000 00)

Bit	Bit Name	Initial Value	R/W	Description
11	UDFDn3	0	R/(W)*	Underflow Flag Dnm These bits indicate whether or not the timer down counter (DCNTDnm) has underflowed. This bit is set to 1 when DCNTDnm is to be decremented while it is H'000000. DCNTDnm holds H'000000 on underflow. Writing H'FFFFFF to DCNTDnm has no effect on these bits. These flags cannot be set to 1 by software. These bits in subblocks D0 to D3 (16 channels) can be cleared by the ACK signal from the DMAC. These bits are initialized to 0 by a reset. Although DCNTDnm is initialized to H'000000, these flags do not indicate underflow because DCNTDnm has not started. For details on control of DCNTDnm, see descriptions of timer D down counters. 0: DCNTDnm has not underflowed [Clearing conditions] • When writing 0 to these bits after reading them as 1 • When DCNTDnm is cleared by an assertion of the ACK signal from the DMAC (supported only by subblocks 0 to 3) 1: DCNTDnm has underflowed [Setting condition] • When DCNTDnm underflowed (decremented while DCNTDnm = H'0000 00)
10	UDFDn2	0	R/(W)*	
9	UDFDn1	0	R/(W)*	
8	UDFDn0	0	R/(W)*	

Bit	Bit Name	Initial Value	R/W	Description
7	CMFADn3	0	R/(W)*	Compare Match A Flag Dnm
6	CMFADn2	0	R/(W)*	These bits indicate whether or not compare match between the output compare register (OCRDnm) and TCNT1Dn has occurred. This flag cannot be set to 1 by software.
5	CMFADn1	0	R/(W)*	
4	CMFADn0	0	R/(W)*	When operation of compare match between OCRDnm and TCNT1Dn is enabled by the setting in timer I/O control register 1 (TIOR1Dn), compare match operation is performed regardless of the state of TCNT1Dn. These bits are set to 1 on the first edge of the Pϕ clock after the values in TCNT1Dn and OCRDnm match, Even if these compare match flags are cleared to 0 by software while TCNT1Dn = OCRDnm after the compare match is detected, these bits are not set to 1 again. A single pulse, signaling detection of compare match A, whose width is equal to the cycle of the Pϕ clock is output to activate the A/D converter (supported by 8 channels in subblocks D0 and D1). To clear these bits, write 0 to these bits after reading them as 1. If TCNT1Dn matches OCRDnm again before the status flag is cleared, the compare match A is detected and the status flag is rewritten with 1. An interrupt request indicating compare match A can be output by setting the compare match A enable bit (CMEADn) in timer interrupt enable register Dn (TIERDn) to 1. The signal line is shared with the compare match B flag. The interrupt signal is asserted either when the compare match A flag and compare match A enable bit are both set to 1 or when the compare match B flag and compare match B enable bit are both set to 1. 0: Compare match A has not occurred [Clearing condition] • When writing 0 to these bits after reading them as 1 1: Compare match A has occurred [Setting condition] • When the values in timer counter 1 (TCNT1Dn) and the output compare register (OCRDnm) match while operation of compare match with OCRDnm is enabled

Bit	Bit Name	Initial Value	R/W	Description
3	CMFBDn3	0	R/(W)*	Compare Match B Flag Dnm These bits indicate whether or not compare match between the general register (GRDnm) and TCNT2Dn has occurred. This flag cannot be set to 1 by software. When operation of compare match between GRDnm and TCNT2Dn is enabled by the setting in timer I/O control register 2 (TIOR2Dn), compare match operation is performed regardless of the state of TCNT2Dn. These bits are set to 1 on the first edge of the Pϕ clock after the values in TCNT2Dn and GRDnm match. Even if these compare match flags are cleared to 0 by software while TCNT2Dn = GRDnm after the compare match is detected, these bits are not set to 1 again. A single pulse, signaling detection of compare match B, whose width is equal to the cycle of the Pϕ clock is output to activate the A/D converter (supported by 8 channels in subblocks D0 and D1). To clear these bits, write 0 to these bits after reading them as 1. If TCNT2Dn matches GRDnm again before the status flag is cleared, the compare match B is detected and the status flag is rewritten with 1. An interrupt request indicating compare match B can be output by setting the compare match B enable bit (CMEBDn) in timer interrupt enable register Dn (TIERDn) to 1. The signal line is shared with the compare match A flag. The interrupt signal is asserted either when the compare match A flag and compare match A enable bit are both set to 1 or when the compare match B flag and compare match B enable bit are both set to 1. 0: Compare match B has not occurred [Clearing condition] • When writing 0 to these bits after reading them as 1 1: Compare match B has occurred [Setting condition] • When the values in timer counter 2 (TCNT2Dn) and general register (GRDnm) match while operation of compare match with GRDnm is enabled
2	CMFBDn2	0	R/(W)*	
1	CMFBDn1	0	R/(W)*	
0	CMFBDn0	0	R/(W)*	

Notes: n = 0 to 3 (n denotes subblock number), m = 0 to 3 (m denotes channel number).

* Only 0 can be written to this bit after it is read as 1 to clear the flag. Writing 1 to this bit is ignored.

13.19.9 Timer Interrupt Enable Registers D0 to D3 (TIERD0 to TIERD3)

TIERD0 to TIERD3 are 16-bit readable/writable registers that enable and disable interrupt requests for overflow on two timer counters (TCNT2Dn and TCNT1Dn) and compare matches between TCNT1Dn and the output compare register (OCRDnm) and between TCNT2Dn and general register (GRDnm).

TIERD0 to TIERD3 can be read from and written to in byte- or word-units.

TIERD0 to TIERD3 are initialized to H'0000 by a power-on reset or a transition to the hardware standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	OVE2 Dn	OVE1 Dn	UDE Dn3	UDE Dn2	UDE Dn1	UDE Dn0	CME ADn3	CME ADn2	CME ADn1	CME ADn0	CME BDn3	CME BDn2	CME BDn1	CME BDn0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: n = 0 to 3 (correspond to subblocks D0 to D3)

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	OVE2Dn	0	R/W	Overflow Interrupt Enable 2Dn Enables and disables an interrupt request when timer counter 2Dn (TCNT2Dn) has overflowed. By setting this bit, overflow flag 2 (OVF2Dn) in timer status register (TSRDn) can be used as a interrupt request source. 0: Interrupt request by OVF2Dn is disabled 1: Interrupt request by OVF2Dn is enabled
12	OVE1Dn	0	R/W	Overflow Interrupt Enable 1Dn Enables and disables an interrupt request when timer counter 1Dn (TCNT1Dn) has overflowed. By setting this bit, overflow flag 1 (OVF1Dn) in timer status register (TSRDn) can be used as a interrupt request source. 0: Interrupt request by OVF1D is disabled 1: Interrupt request by OVF1D is enabled

Bit	Bit Name	Initial Value	R/W	Description
11	UDEDn3	0	R/W	Underflow Interrupt Enable Dnm These bits enable and disable an interrupt request when timer down counter Dnm (DCNTDnm) has underflowed. By setting these bits, underflow flag (UDFDnm) in timer status register (TSRDn) can be used as a interrupt request source. 0: Interrupt request by UDFDnm is disabled 1: Interrupt request by UDFDnm is enabled
10	UDEDn2	0	R/W	
9	UDEDn1	0	R/W	
8	UDEDn0	0	R/W	
7	CMEADn3	0	R/W	Compare Match Interrupt Enable Dnm These bits enable and disable an interrupt request when compare match between output compare register (OCRDnm) and TCNT1Dn has occurred. By setting these bits, compare match A flag (CMFADn) in timer status register (TSRDn) can be used as a interrupt request source. However, the signal line is shared with compare match B. 0: Interrupt request by CMFADnm is disabled 1: Interrupt request by CMFADnm is enabled
6	CMEADn2	0	R/W	
5	CMEADn1	0	R/W	
4	CMEADn0	0	R/W	
3	CMEBDn3	0	R/W	Compare Match Interrupt Enable Dnm These bits enable and disable an interrupt request when compare match between output compare register (GRDnm) and TCNT2Dn has occurred. By setting these bits, compare match B flag (CMFBDn) in timer status register (TSRDn) can be used as a interrupt request source. However, the signal line is shared with compare match A. 0: Interrupt request by CMFBDnm is disabled 1: Interrupt request by CMFBDnm is enabled
2	CMEBDn2	0	R/W	
1	CMEBDn1	0	R/W	
0	CMEBDn0	0	R/W	

Note: n = 0 to 3 (n denotes subblock number), m = 0 to 3 (m denotes channel number).

13.19.10 Compare Match Pulse Output Control Registers D0 and D1 (CMPOD0 and CMPOD1)

CMPOD0 and CMPOD1 are 8-bit readable/writable registers that select whether or not a pulse to activate the A/D converter is output on compare matches A and B.

CMPOD0 and CMPOD1 can be read from and written to in byte or word units.

CMPOD0 and CMPOD1 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	CMP BDn3	CMP BDn2	CMP BDn1	CMP BDn0	CMP ADn3	CMP ADn2	CMP ADn1	CMP ADn0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: n = 0, 1 (correspond to subblocks D0 and D1)

Bit	Bit Name	Initial Value	R/W	Description
7	CMPBDn3	0	R/W	Compare Match B Pulse Output Control These bits select whether or not a pulse is output on compare match B. When these bits are set to 1, an active-low pulse whose width is equal to the cycle of the Pϕ clock is output on compare match B. 0: Compare match B pulse is not output 1: Compare match B pulse is output (pulse width = cycle of the Pϕ clock)
6	CMPBDn2	0	R/W	
5	CMPBDn1	0	R/W	
4	CMPBDn0	0	R/W	
3	CMPADn3	0	R/W	Compare Match A Pulse Output Control These bits select whether or not a pulse is output on compare match A. When these bits are set to 1, an active-low pulse whose width is equal to the cycle of the Pϕ clock is output on compare match A. 0: Compare match A pulse is not output 1: Compare match A pulse is output (pulse width = cycle of the Pϕ clock)
2	CMPADn2	0	R/W	
1	CMPADn1	0	R/W	
0	CMPADn0	0	R/W	

Note: n = 0 to 3 (n denotes subblock number)

13.19.11 Timer Output Control Registers D0 to D3 (TOCRD0 to TOCRD3)

TOCRD0 to TOCRD3 are 8-bit readable/writable registers that select whether or not signals on output pins for subblock Dn (TODnmA and TODnmB) are inverted.

Signals on pins TODnmA and TODnmB are inverted on the first edge of the Pϕ clock after the output inversion select Dn bit is set to 1. This function is not affected by the operating state of timer counters 1Dn and 2Dn (TCNT1Dn, TCNT2Dn).

The TONEBDn bit controls four outputs (TODn0B, TODn1B, TODn2B, TODn3B) in a single subblock. The TONEADn bit controls four outputs (TODn0A, TODn1A, TODn2A, TODn3A) in a single subblock. These bits cannot control individual signals independently (channel control is not available).

Output levels on pins TODnmA and TODnmB are initialized to a level of 0 (when TONEADn = 0 and TONEBDn = 0).

TOCRD0 to TOCRD3 can be read from and written to in byte or word units.

TOCRD0 to TOCRD3 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	TONE BDn	TONE ADn
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W

Note: n = 0 to 3 (correspond to subblocks D0 to D3)

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	TONEBDn	0	R/W	Output Inversion Select TODnmB Selects whether or not the output level on pin TODnmB is inverted 0: Output level is not inverted 1: Output level is inverted

Bit	Bit Name	Initial Value	R/W	Description
0	TONEADn	0	R/W	Output Inversion Select TODnmA Selects whether or not the output level on pin TODnmA is inverted 0: Output level is not inverted 1: Output level is inverted

Note: n = 0 to 3 (n denotes subblock number), m = 0 to 3 (m denotes channel number)

13.19.12 Timer Offset Base Registers D0 to D3 (OSBRD0 to OSBRD3)

OSBRD0 to OSBRD3 are 32-bit read-only registers that are used only for capture. The value in timer counter 1Dn (TCNT1Dn) is captured in these registers by a trigger signal from timer A. Pin TIA01 or TIA02 can be selected as a trigger. Trigger signals for subblocks D0 to D2 are selected by bit EVOSEL2A in TCRA and trigger signals for subblock D3 are selected by bit EVOSEL2B in TCRA. For details, see section 13.10.1, Timer Control Register A (TCRA).

OSBRD0 to OSBRD3 can be read from in longword units.

OSBRD0 to OSBRD3 are initialized to H'00000000 by a power-on reset or a transition to the hardware standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

13.19.13 Timer Counter 1D0 to 1D3 (TCNT1D0 to TCNT1D3)

TCNT1D0 to TCNT1D3 are 32-bit readable/writable registers driven by the clock selected in the CKSEL1Dn[2:0] bits in timer control register Dn (TCRDn). These counters are started by setting the bit in timer start register (TSTRD) to 1.

When the counter overflows, the over flag (OVF1Dn) in timer status register Dn (TSRDn) is set to 1.

TCNT1D0 to TCNT1D3 can be read from and written to in longword units.

TCNT1D0 to TCNT1D3 is initialized to H'0000 0000 by a power-on reset or a transition to the hardware standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

13.19.14 Timer Counters 2D0 to 2D3 (TCNT2D0 to TCNT2D3)

TCNT2D0 to TCNT2D3 are 32-bit readable/writable registers driven by the clock selected in the CKSEL2Dn[2:0] bits in timer control register Dn (TCRDn). These counters are started by setting the bit in the timer start register Dn (TSTRD) to 1.

When the counter overflows, the over flag (OVF2Dn) in timer status register Dn (TSRDn) is set to 1.

TCNT2D0 to TCNT2D3 can be read from and written to in longword units.

TCNT2D0 to TCNT2D3 is initialized to H'00000000 by a power-on reset or a transition to the hardware standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

13.19.15 Output Compare Registers D00 to D33 (OCRD00 to OCRD33)

OCRD00 to OCRD33 are 32-bit readable/writable registers. The upper 24 bits are available.

OCRDnm is constantly compared with the up counter TCNT1Dn. When compare match operation is selected by bit IOADnm in TIOR1Dn, the compare match A flag (CMFADnm) in TSRDn is set to 1 on the first edge of the Pϕ clock after the values in TCNT1Dn and OCRDnm match. When compare match A is selected by an output source select bit, a signal is output to pin TODnmA on compare match.

When compare match A is selected as a down counter starting trigger by the TRGSELDnm bit in DCRDnm, DCNTDnm is ready to be counted down on compare match A.

When the down counter (DCNTDnm) is ready, it is started in synchronization with the down counter clock. At this time, a one-shot pulse can be output on pin TODnmB. If compare match A and down-counter stop trigger are output at the same time, output is disabled without any pulse.

When TCNT1Dn overflows to change from H'FFFFFF to H'000000 and OCRDnm is set to H'000000, compare match is detected.

By setting bit CMEADn in timer interrupt enable register Dn (TIERDn) to 1, an interrupt request can be issued on compare match A. (The output line is shared with compare match B. Confirm that which interrupt is occurred by TSRDn).

OCRD00 to OCRD33 can be read from and written to in longword units.

OCRD00 to OCRD33 are initialized to H'FFFFFF00 by a power-on reset or a transition to the hardware standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									-	-	-	-	-	-	-	-
Initial value:	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

13.19.16 General Registers D00 to D33 (GRD00 to GRD33)

GRD00 to GRD33 are 32-bit readable/writable registers that function as the capture register and compare match register. The functions are switched by timer I/O control register 2Dn (TIOR2Dn).

For the capture function, these registers capture the value in TCNT2Dn on compare match A. Even if TCNT2Dn is stopped (the TDE bit in ATUENR is cleared to 0 or the STRDn bit in TSTRD is cleared to 0), capture continues and the value in TCNT2Dn stopped is captured by GRDnm.

For the compare match function, these registers are constantly compared with TCNT2Dn. When compare match is enabled by the IOBDnm bit in TIOR2Dn, the CMFBDnm bit in TSRDn is set to 1 in synchronization with the Pϕ clock after the values in TCNT2Dn and GRDnm match. When compare match B is selected as the output source by the OSSDnm bit, a signal is output on pin TODnmA on compare match.

When compare match B is selected as the down counter starting trigger by the TRGSELDnm bit in DCRDnm, DCNTDnm is ready to be counted down on compare match B.

When DCNTDnm is ready, it is decremented in synchronization with the down counter clock. At this time, a one-shot pulse can be output on pin TODnmB.

When compare match B is selected as the down counter stopping trigger by the TRGSELDnm bit in DCRDnm, DCNTDnm is not ready for counting down. The counter is cleared to 0 and the output signal on pin TODnmB is negated (output of one-shot pulse is terminated) in synchronization with the down counter clock after it is not ready because of compare match B.

When TCNT2Dn overflows to change from H'FFFFFF to H'000000 and GRDnm is set to H'000000, compare match is detected.

By setting bit CMEBDn in timer interrupt enable register Dn (TIERDn) to 1, an interrupt request can be issued on compare match B. (The output line is shared with compare match A. Confirm that which interrupt is occurred by TSRDn).

GRD00 to GRD33 can be read from and written to in longword units.

GRD00 to GRD33 are initialized to H'FFFF FF00 by a power-on reset or a transition to the hardware standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									-	-	-	-	-	-	-	-
Initial value:	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

13.19.17 Timer Down Counters D00 to 33 (DCNTD00 to DCNTD33)

DCNTD00 to DCNTD33 are 32-bit readable/writable registers driven by the clock selected in the DCSELDn[2:0] bit in timer control register Dn (TCRDn).

DCNTDnm is controlled by the down counter control register Dn (DCRDn). This register starts to be decremented; when compare match A or B is detected; when 1 is written to the DSTDnm bit in DSTRDn. Counting is stopped when DCNTDnm underflows or on compare match B. While the down counter is enabled, it is decremented every input of the down counter clock.

Decrementation is enabled; on the first edge of the Pϕ clock (the same as compare match A) after the values in TCNT1Dn and OCRDnm match; on the first edge of the Pϕ clock (the same as compare match B) after the values in TCNT2Dn and GRDnm match; on the first edge of the Pϕ clock after the DSTDnm bit is set to 1. Decrementation is enabled until DCNTDnm underflows or until the first edge of the Pϕ clock (the same as compare match B) after the values in TCNT2Dn and GRDnm match. The down counter is decremented every input of the down counter clock while it is enabled.

Once DCNTDnm is enabled, it remains enabled until DCNTDnm underflows or on compare match B (when counter stopping trigger is selected). While it is enabled, another counter starting trigger or writing 1 to DSTDnm bit has no effect on the enabled state of the counter.

The counter is stopped on the first edge of the down counter clock after compare match B if the counter stopping trigger is selected and then is cleared to H'000000. When the counter is started or stopped by compare match A and writing 1 to DSTDnm simultaneously, counter stopping trigger takes priority. The counter is not decremented and no signal on pin TODnmB is output. Moreover, when no down counter clock is input during the enabled state, the counter is not decremented. The counter which is stopped does not change the value. If a value other than H'000000 is set in DCNTDnm after decrementation is terminated by underflow, the counter is not decremented until the counter stopping source is activated.

DCNTD00 to DCNTD33 can be read from and written to in longword units.

DCNTD00 to DCNTD33 are initialized to H'00000000 by a power-on reset or a transition to the hardware standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

13.20 Operations of Timer D

One-shot pulse can be output from timer D. By using compare match A or B as a start trigger of a down counter, one-shot pulse with an offset can also be output.

Setting the TDE bit in ATUENR and bit STRDn in timer start register (TSTRD) to 1 makes two up-counters (TCNT1Dn and TCNT2Dn) in subblock Dn start operation.

The down counter is started by setting bit DSTDnm in down counter starting register D (DSTRD) to 1, compare match A, or compare match B. Compare matches A and B are selected by bit TRGSELnm in down counter control register (DCRD). When any of these triggers occurs, DCNTDnm driven by the down counter clock is started.

The down counter is stopped on an underflow of the down counter and compare match B selected by bit TRFSELmn. The down counter is stopped immediately after it underflows (the counter is to be decremented when the value is H'000000). It is stopped and cleared to H'000000 on the first edge of the down counter clock after the trigger is detected.

For compare match A between TCNT1Dn and OCRDnm and compare match B between TCNT2Dn and GRDnm, the CMFADnm and CMFBDnm bits in TSRDn are set to 1 on the first edge of the Pϕ clock after the compare match.

A signal on pin TODnmA is output when the output source selected in the OSSDnm bit in timer I/O control register 1D (TIOR1Dn) is activated. For example, assume that compare match A is selected. A signal level set by the IOAnm bit is output on pin TODnmA on the first edge of the Pϕ clock after the compare match between TCNT1Dn and OCRDnm.

Output of the one-shot pulse is synchronized with the down counter clock in a way similar to the down counter operation. Since the three counter starting sources are synchronized with the Pϕ clock, a signal on pin TODnmB is output on the first edge of the down counter clock after the source is activated. Underflow of the down counter, which is a counter stopping source, is synchronized with the down counter clock and compare match B is synchronized with the Pϕ clock. As to negation timing, the TODnmB signal is negated in synchronization with underflow and on the first edge of the down counter clock after the compare match B.

Table 13.17 Output Timing for One-Shot Pulse (TODnmB)

Output Pin	Assertion Timing	Negation Timing	Initial Value
TODnmB	On the first edge of the down counter clock after the counter starting source is activated	On the first edge of the down counter clock after compare match or on DCNTDnm underflow	0 (inverted depending on TOCRDn)

Note: If an assertion and a negation occur simultaneously, the negation takes priority.

The initial values output on pins TODnmA and TODnmB are 0. However, the output level can be inverted by setting output control register Dn (TOCRDn). Setting bit TONEADn to 1 makes pin TODnmA in subblock Dn inverted and setting bit TONEBDn to 1 makes pin TODnmB in subblock Dn inverted.

An interrupt request can be output by setting bits CMEADnm and CMEBDnm to 1 using compare match A or B flag. The signal line is shared with compare matches A and B. When the status flag and enable bit for each interrupt are both set to 1, the output level of the interrupt request is asserted (active-low signal).

To set the clocks for TCNT1Dn, TCNT2Dn, or DCNTDnm or values in registers such as DCNTDnm, OCRDnm, and GRDnm while TCNT1Dn or TCNT2Dn is in operation, note that the value to be set may lead to malfunction. For example, while setting the compare match value, the counter value may exceed the value to be set.

Figure 13.28 shows an operation example of one-shot pulse output for channel 0 in subblock D0.

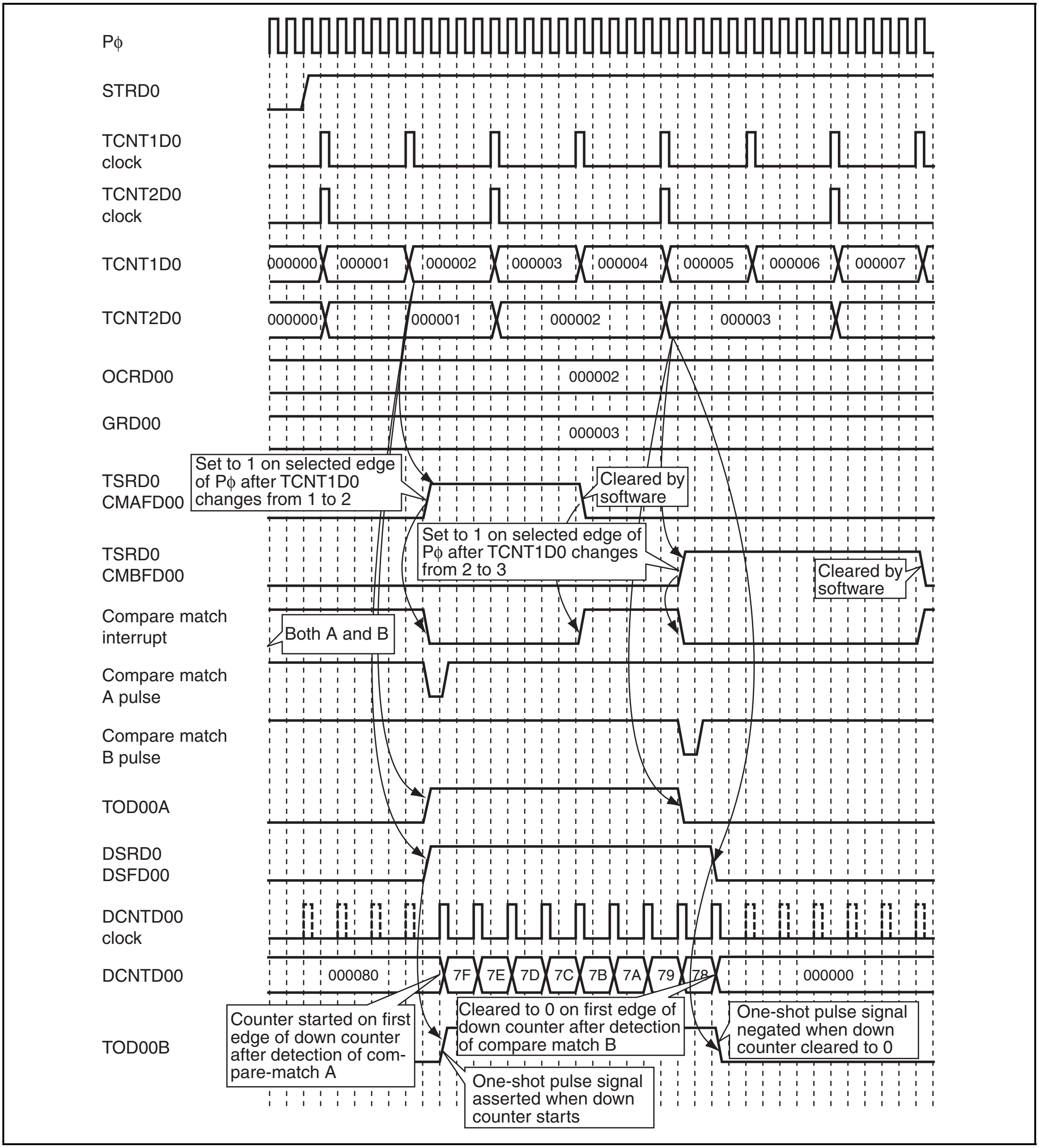

Pϕ
STRD0
TCNT1D0 clock
TCNT2D0 clock
TCNT1D0
000000
000001
000002
000003
000004
000005
000006
000007
TCNT2D0
000000
000001
000002
000003
OCRD00
000002
GRD00
000003
TSRD0 CMAFD00
Set to 1 on selected edge of Pϕ after TCNT1D0 changes from 1 to 2
Cleared by software
TSRD0 CMBFD00
Set to 1 on selected edge of Pϕ after TCNT1D0 changes from 2 to 3
Cleared by software
Compare match interrupt
Both A and B
Compare match A pulse
Compare match B pulse
TOD00A
DSRD0 DSFD00
DCNTD00 clock
DCNTD00
000080
7F
7E
7D
7C
7B
7A
79
78
000000
TOD00B
Counter started on first edge of down counter after detection of com-pare-match A
Cleared to 0 on first edge of down counter after detection of compare match B
One-shot pulse signal negated when down counter cleared to 0
One-shot pulse signal asserted when down counter starts

Figure 13.28 Operation Example of One-Shot Pulse Output (1)
(Counting Started on Compare Match A and Stopped on Compare Match B)

Figure 13.28 shows operations of the down counter when the counter is started on compare match A and is stopped on compare match B. It also shows the assertion and negation of a one-shot pulse. Compare match A as the counter starting trigger and compare match B as the counter stopping trigger are set by the TRGSELD00 bits in DCRD0. Both matches are selected as the source of the output signal by the OSSD00 bits in TIOR1D0 and output levels are set by the IOAD00 and IOBD00 bits. A logical zero for compare match A and a logical one for compare match A are selected.

TCNT1D0 and TCNT2D0 are started for counting up on the first edge of the counter clock after the counter Dn start bit in timer start register (TSTRD) is set to 1. The compare match A status flag (CMFAD00) is set to 1 on the first edge of the Pϕ clock after the values in TCNT1D0 and the output compare register (OCRD00) match. At the same time, a level of 1 is output on pin TOD00A and the down counter status flag (DSFD00) is set to 1 to make the down counter ready for counting down. The down counter keeps the ready state until compare match B or DCNTD00 underflow. DCNTD00 is started by the edge of the down counter clock. At this time, a level of 1 is output on pin TOD00B.

The compare match B flag (CMFBD00) is set to 1 on the first edge of the Pϕ clock after the values in GRD00 and TCNT2D0 match. At this time, a level of 0 is output on pin TOD00A. The down counter is cleared and a one-shot pulse (TOD00B) is terminated.

An interrupt request can be issued on compare matches A and B. For compare match A, set the CMEAD00 bit in timer interrupt enable register D0 (TIERED0) and for compare match B, set the CMEBD00 bit. Since the signal line is shared with both matches, read the CMAFD00 and CMBFD00 bits in TSRD0 to know the interrupt source.

Moreover, a pulse for A/D activation can be output by setting compare match pulse output control register D0 (CMPOD0) at the same time as the interrupt request to be output. The pulse width is equal to the cycle of the Pϕ clock.

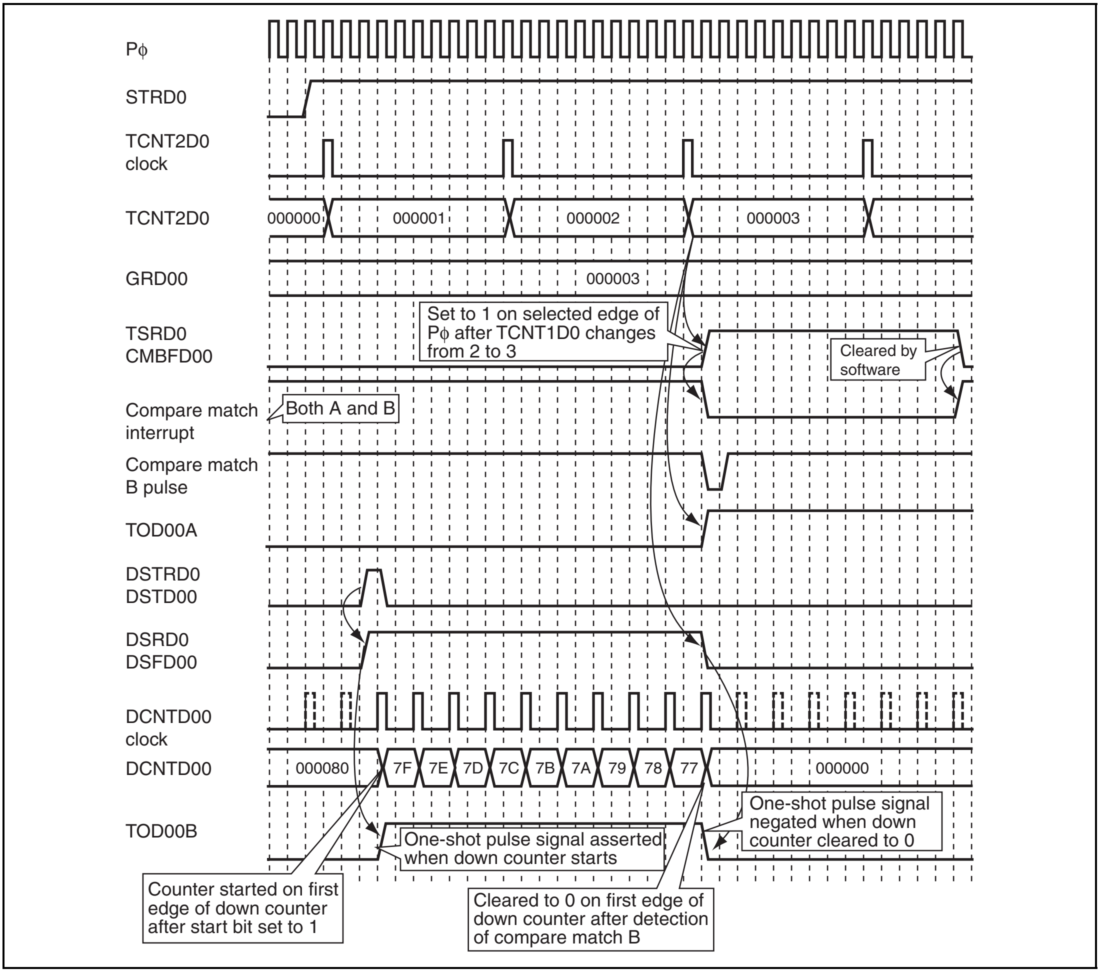

Pϕ
STRD0
TCNT2D0 clock
TCNT2D0
000000
000001
000002
000003
GRD00
000003
TSRD0 CMBFD00
Set to 1 on selected edge of Pϕ after TCNT1D0 changes from 2 to 3
Cleared by software
Compare match interrupt
Both A and B
Compare match B pulse
TOD00A
DSTRD0 DSTD00
DSRD0 DSFD00
DCNTD00 clock
DCNTD00
000080
7F
7E
7D
7C
7B
7A
79
78
77
000000
TOD00B
One-shot pulse signal asserted when down counter starts
One-shot pulse signal negated when down counter cleared to 0
Counter started on first edge of down counter after start bit set to 1
Cleared to 0 on first edge of down counter after detection of compare match B

Figure 13.29 Operation Example of One-Shot Pulse Output (2)
(Counting Started by Writing 1 to Counter Starting Bit and Stopped on Compare Match B)

Figure 13.29 shows an operation example when the down counter is started by writing 1 to the down counter starting bit. In this example, the counter starting trigger is not selected and compare match B is selected as the counter stopping trigger (TRGSELD00 in DCRD0). The source of the output signal is compare match B (OSSD00 in TIOR1D0) and logical one is output (IOBD00 in TIOR1D0).

Figure 13.30 shows an operation example of a one-shot pulse output for channel 0 in subblock D0. Setting the DSTD00 bit in the down counter starting register (DSTRD0) sets the down counter status flag (DSFD00) to 1. This makes the down counter ready for counting down. DCNTD00 is started on the first edge of the down counter clock after DSFD00 is set to 1. At this time, a level of 1 is output on pin TOD00B.

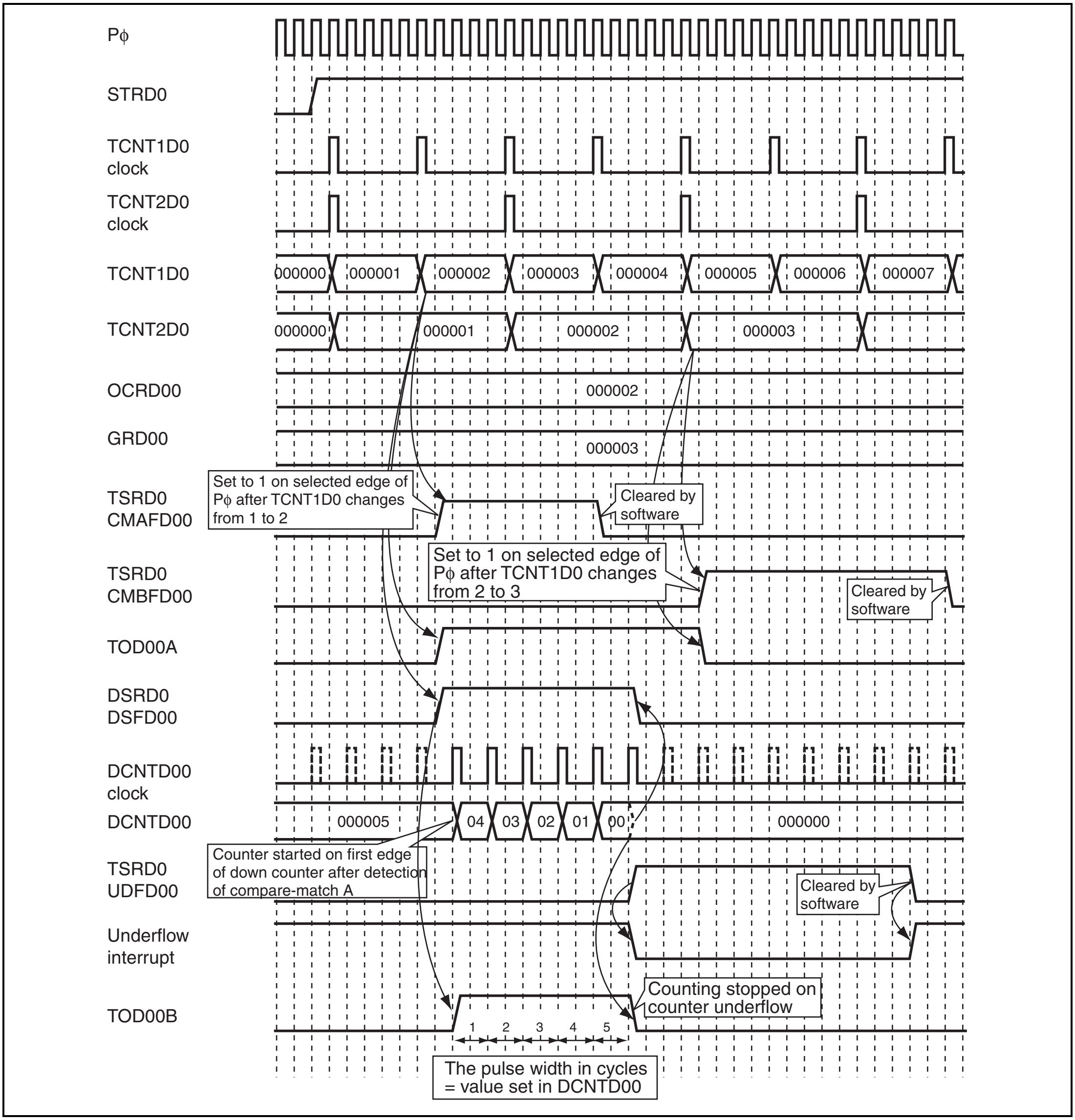


Figure 13.30 Operation Example of One-Shot Pulse Output (3) (Underflow Occurs)

Figure 13.30 shows an operation example when a one-shot pulse is terminated on underflow. In this example, compare match A as the counter starting trigger is selected and the counter stopping trigger is not selected (TRGSELD00 in DCRD0). Both compare matches A and B as the source of the output signal (OSSD00 in TIOR1D0) are selected. Output levels of 0 for compare match A (IOAD00) and 1 for compare match B (IOBD00) are selected.

Underflow is detected and the underflow flag (UDFD00) in TSRD0 is set on the first edge of the down counter clock after the value in timer down counter D00 (DCNTD00) is H'000000. At the same time, the one-shot pulse output is terminated. The width of the pulse output on pin TOD00B is equal to the value set in DCNTD00 before counting down.

13.21 Overview of Timer E

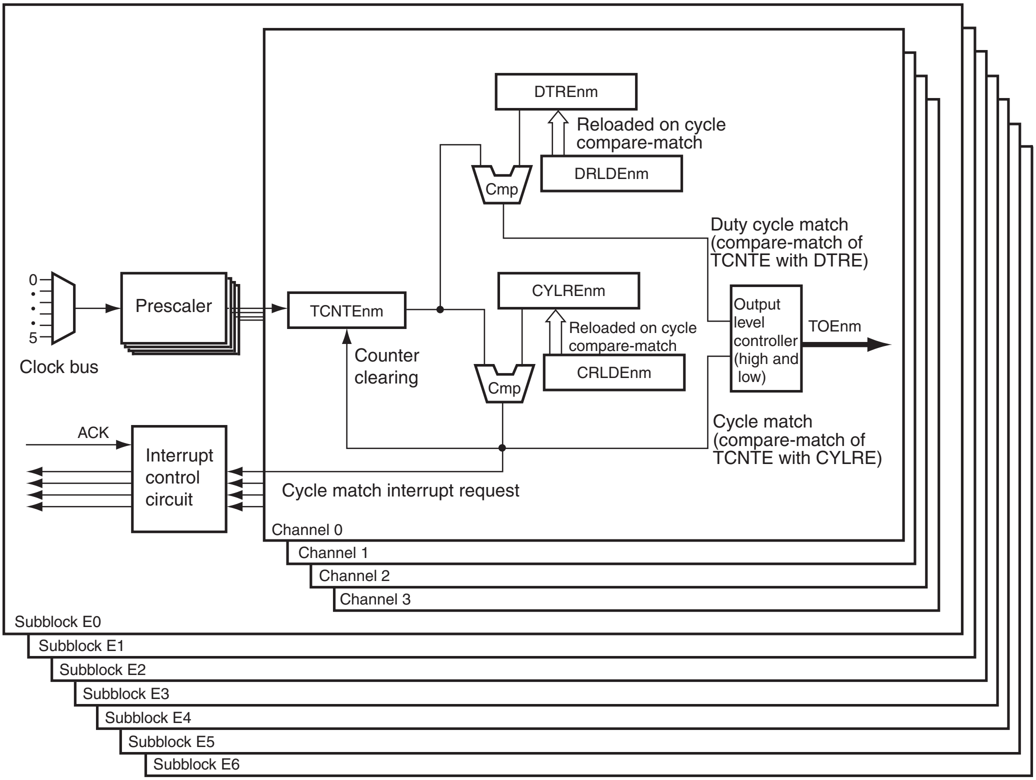
Timer E consists of seven subblocks that generate PWM outputs. The subblocks are identical each other. Timer E has the following functions.

- Output of waveform with a duty cycle of 0 to 100% by setting cycle-setting register and duty cycle-setting register
- The values of the cycle-setting and duty-cycle-setting registers are updated every PWM cycle. The values in the cycle reload register and duty cycle reload register are reloaded as update data. The reloading function can be enabled and disabled.
- Forcible termination of PWM cycle by writing H'0000 to the counter
- On-state duty (active-high output) and off-state duty (active-low output) modes available
- Interrupt requests can be issued on cycle match (compare match between cycle setting register and timer counter), that is, interrupts are issued every cycle.
- Cycle matches of channel 0 can be used as DMAC activation interrupts, which are automatically cleared by the ACK signal.

13.21.1 Block Diagram of Timer E

Timer E consists of seven subblocks. Each subblock includes four channels. Each channel has timer counter E (TCNTE), cycle-setting register E (CYLRE), duty cycle setting register E (DTRE), cycle reload register E (CRLDE), duty cycle reload register E (DRLDE), and controller. Each channel has an output pin for PWM waveforms.

Figure 13.31 is a block diagram of timer E.


DTREnm
Reloaded on cycle compare-match
DRLDEnm
Cmp
Duty cycle match (compare-match of TCNTE with DTRE)
0
5
Clock bus
Prescaler
TCNTEnm
CYLREnm
Reloaded on cycle compare-match
CRLDEnm
Cmp
Output level controller (high and low)
TOEnm
Counter clearing
Cycle match (compare-match of TCNTE with CYLRE)
ACK
Interrupt control circuit
Cycle match interrupt request
Channel 0
Channel 1
Channel 2
Channel 3
Subblock E0
Subblock E1
Subblock E2
Subblock E3
Subblock E4
Subblock E5
Subblock E6

Note: n = 0 to 6 (n denotes subblock number), m = 0 to 3 (m denotes channel number)

[Legend]

TCNTE:	Timer counter E
DTRE:	Duty cycle register E
DRLDE:	Duty cycle reloading register E
CYLRE:	Cycle register E
CRLDE:	Cycle reloading register E

Figure 13.31 Block Diagram of Timer E

13.22 Description of Timer E Registers

13.22.1 Timer Start Register E (TSTRE)

TSTRE is an 8-bit readable/writable register that controls subblocks E0 to E6.

The timer E counters run when the timer E enable bit (TEE) in the ATU-III master enable register (ATUENR), timer start register E (TSTRE), and subblock starting register E (SSTRE) must be set.

TSTRE can be read from and written to in byte or word units.

TSTRE is initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	-	STRE6	STRE5	STRE4	STRE3	STRE2	STRE1	STRE0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6	STRE6	0	R/W	Subblock E6 Start 0: Subblock E6 is disabled 1: Subblock E6 is enabled
5	STRE5	0	R/W	Subblock E5 Start 0: Subblock E5 is disabled 1: Subblock E5 is enabled
4	STRE4	0	R/W	Subblock E4 Start 0: Subblock E4 is disabled 1: Subblock E4 is enabled
3	STRE3	0	R/W	Subblock E3 Start 0: Subblock E3 is disabled 1: Subblock E3 is enabled

Bit	Bit Name	Initial Value	R/W	Description
2	STRE2	0	R/W	Subblock E2 Start 0: Subblock E2 is disabled 1: Subblock E2 is enabled
1	STRE1	0	R/W	Subblock E1 Start 0: Subblock E1 is disabled 1: Subblock E1 is enabled
0	STRE0	0	R/W	Subblock E0 Start 0: Subblock E0 is disabled 1: Subblock E0 is enabled

13.22.2 Subblock Starting Registers E0 to E6 (SSTRE0 to SSTRE6)

SSTRE0 to SSTRE6 are 8-bit readable/writable registers that enable and disable the timer counters for four channels of a subblock. Subblocks selected by timer start register E (TSTRE) is enabled. However, both the SSTRE and TEE bits must be set to start counting.

The prescalers run regardless of the counter Enm start bit and are not synchronized with the timing at which TCNTE is started. Therefore, the time from when the counter Enm start bit is set to when TCNTE is incremented for the first time is less than the cycle of the clock of TCNTE.

SSTRE0 to SSTRE6 can be read from and written to in byte or word units.

SSTRE0 to SSTRE6 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	SSTREn3	SSTREn2	SSTREn1	SSTREn0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

Note: n = 0 to 6 (correspond to subblocks E0 to E6)

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	SSTREn3	0	R/W	Counter En3 Start Enables and disables timer counter En3 (TCNTEn3). When this bit is cleared to 0, TCNTEn3 is disabled. TCNTEn3 retains the previous value while it is stopped. When this bit is set to 1, TCNTEn3 is resumed from the retained value. 0: Counter of channel 3 in subblock En is disabled 1: Counter of channel 3 in subblock En is enabled
2	SSTREn2	0	R/W	Counter En2 Start Enables and disables timer counter En2 (TCNTEn2). When this bit is cleared to 0, TCNTEn2 is disabled. TCNTEn2 retains the previous value while it is stopped. When this bit is set to 1, TCNTEn2 is resumed from the retained value. 0: Counter of channel 2 in subblock En is disabled 1: Counter of channel 2 in subblock En is enabled
1	SSTREn1	0	R/W	Counter En1 Start Enables and disables timer counter En1 (TCNTEn1). When this bit is cleared to 0, TCNTEn1 is disabled. TCNTEn3 retains the previous value while it is stopped. When this bit is set to 1, TCNTEn1 is resumed from the retained value. 0: Counter of channel 1 in subblock En is disabled 1: Counter of channel 1 in subblock En is enabled
0	SSTREn0	0	R/W	Counter En0 Start Enables and disables timer counter En0 (TCNTEn0). When this bit is cleared to 0, TCNTEn0 is disabled. TCNTEn0 retains the previous value while it is stopped. When this bit is set to 1, TCNTEn0 is resumed from the retained value. 0: Counter of channel 0 in subblock En is disabled 1: Counter of channel 0 in subblock En is enabled

Note: n = 0 to 6 (n denotes subblock number)

13.22.3 Prescaler Registers E0 to E6 (PSCRE0 to PSCRE6)

PSCRE0 to PSCRE6 are 8-bit readable/writable registers. Each subblock of timer E has one prescaler that divides the frequency of the clock supplied via the clock bus. The register sets the division ratio of the prescalers.

When the value in prescaler register E (PSCRE) is changed, the prescaler updates the value on its underflow. Timer counter E (TCNTE) in the same block is driven by the clock output from prescaler E.

The settable value in prescaler register E (PSCRE) ranges from H'0 to H'7. The division ratio is given below.

$$\text{Division ratio of prescaler} = \frac{1}{\text{PSCEn[2:0]} + 1} \text{ (Settable value: 1/1 to 1/8)}$$

A duty cycle of 50% for the prescaler E output clock is not guaranteed. The high level width is equal to the cycle of the Pϕ clock and a low level is output in the remaining cycle of the prescaler E output clock.

Prescaler E runs when the TEE bit in the ATU-III master enable register (ATUENR) and the subblock E start bit (STRE) in timer start register E (TSTRE) are both set to 1.

PSCRE0 to PSCRE6 can be read from and written to in byte or word units.

PSCRE0 to PSCRE6 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	PSCEn[2:0]		
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

Note: n = 0 to 6 (correspond to subblocks E0 to E6)

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	PSCEn [2:0]	000	R/W	Division Ratio These bits store the division ratio of the prescaler.

13.22.4 Prescaler Channel Registers E0 to E6 (PSCCRE00 to PSCCRE63)

PSCCRE00 to PSCCRE63 are 8-bit readable/writable registers. Each channel of timer E has one prescaler that divides the frequency of the clock supplied via the clock bus. The register sets the division ratio of the prescalers.

When the value in prescaler channel register E (PSCCRE) is changed, the prescaler updates the value on its underflow. Timer counter E (TCNTE) in the same channel is driven by the clock output from prescaler E.

The settable value in prescaler channel register E (PSCCRE) ranges from H'00 to H'FF. The division ratio is given below.

$$\text{Division ratio of prescaler} = \frac{1}{\text{PSCCEnm[7:0]} + 1} \text{ (Settable value: 1/1 to 1/256)}$$

A duty cycle of 50% for the prescaler E output clock is not guaranteed. The high level width is equal to the cycle of the Pϕ clock and a low level is output in the remaining cycle of the prescaler E output clock.

Prescaler E runs when the TEE bit in the ATU-III master enable register (ATUENR) and the subblock E start bit (STRE) in timer start register E (TSTRE) are both set to 1.

PSCCRE00 to PSCCRE63 can be read from and written to in byte or word units.

PSCCRE00 to PSCCRE63 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	PSCCEnm[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Notes 1 : n = 0 to 6 (correspond to subblocks E0 to E6)
2 : m = 0 to 3 (correspond to channels 0 to 3)

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	PSCCEnm [7:0]	All 0	R/W	Division Ratio These bits store the division ratio of the prescaler.

13.22.5 Timer Control Register E0 to E6 (TCRE0 to TCRE6)

TCRE0 to TCRE6 are 8-bit readable/writable registers that select the counter clock of prescaler E from clock-bus lines 0 to 6. Timer counter E (TCNTE) is driven by the clock output from prescaler E.

TCRE0 to TCRE6 can be read from and written to in byte or word units.

TCRE0 to TCRE6 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	PSC SEL	-	-	-	-	CKSELEn[2:0]		
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R/W	R/W	R/W

Notes 1: n = 0 to 6 (correspond to subblocks E0 to E6)
2: m = 0 to 3 (correspond to channels 0 to 3)

Bit	Bit Name	Initial Value	R/W	Description
7	PSCSEL	0	R/W	Prescaler Select (PSCSEL) The prescaler can be selected in subblock units or in channel units. This selection makes a difference in the range of division ratios that can be selected for the prescaler. 0: These bits select the prescaler registers En (PSCREn: 3-bit settings for subblock units). 1: These bits select the prescaler channel registers Enm (PSCREnm: 8 bit settings for channel units).
6 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
2 to 0	CKSELEn [2:0]	000	R/W	TCNTEn Clock Select These bits select the counter clock of prescaler E from clock-bus lines 0 to 5. 000: Clock-bus line 0 is selected as counter clock of prescaler E 001: Clock-bus line 1 is selected as counter clock of prescaler E 010: Clock-bus line 2 is selected as counter clock of prescaler E 011: Clock-bus line 3 is selected as counter clock of prescaler E 100: Clock-bus line 4 is selected as counter clock of prescaler E 101: Clock-bus line 5 is selected as counter clock of prescaler E 11x: Reserved

[Legend]

x: Don't care

13.22.6 Reload Control Registers E0 to E6 (RLDCRE0 to RLDCRE6)

RLDCRE0 to RLDCRE6 are 8-bit readable/writable registers that enable and disable the reload function.

RLDCRE0 to RLDCRE6 can be read from and written to in byte or word units.

RLDCRE0 to RLDCRE6 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	RLDENEn3	RLDENEn2	RLDENEn1	RLDENEn0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

Note: n = 0 to 6 (correspond to subblocks E0 to E6)

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	RLDENEn3	0	R/W	Reload Enable Enm These bits enable and disable the function with which the duty cycle-setting and cycle-setting registers is reloaded on cycle match. 0: Reload function on cycle match is disabled 1: Reload function on cycle match is enabled
2	RLDENEn2	0	R/W	
1	RLDENEn1	0	R/W	
0	RLDENEn0	0	R/W	

Note: n = 0 to 6 (n denotes subblock number), m = 0 to 3 (m denotes channel number)

13.22.7 Timer Status Registers E0 to E6 (TSRE0 to TSRE6)

TSRE0 to TSRE6 are 8-bit readable/writable registers that indicate occurrence of a cycle match and TCNTE overflow. The cycle match is a compare match between cycle setting register E (CYLRE) and timer counter E (TCNTE).

TSRE0 to TSRE6 can be read from and written to in byte or word units.

TSRE0 to TSRE6 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	OVFEn3	OVFEn2	OVFEn1	OVFEn0	CMFEn3	CMFEn2	CMFEn1	CMFEn0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: n = 0 to 6 (correspond to subblocks E0 to E6)

* Only 0 can be written to this bit after it is read as 1 to clear it. Writing 1 is ignored.

Bit	Bit Name	Initial Value	R/W	Description
7	OVFEn3	0	R/(W)*	Overflow Flag Enm These bits are set to 1 when timer counter E (TCNTE) overflows. This flag cannot be set to 1 by software. Overflow occurs when the counter is incremented while it is H'FFFF. Writing H'0000 to the counter has no effect on these bits. When writing a value to the counter and incrementation occur simultaneously while the counter is H'FFFF, the overflow flag is set to 1 but the counter value is changed to the written value instead of H'0000. No interrupt corresponds to these flags. As to cycle match, since the counter is cleared to H'0001, overflow will not occur. However, it may occur when the value in the cycle-setting register is changed during counter in operation. When overflow and cycle match occur simultaneously, overflow is not detected (the counter is incremented while it is H'FFFF and CYLREn is H'FFFF). In this case, only the cycle match is handled. If the counter value is H'0001 and the reload function is enabled, cycle reload or duty-cycle reload is performed. 0: Counter E has not overflowed [Clearing condition] • When writing 0 to these bits after reading them as 1 1: Counter E has overflowed [Setting condition] • When counter E value changes from H'FFFF to H'0000
6	OVFEn2	0	R/(W)*	
5	OVFEn1	0	R/(W)*	
4	OVFEn0	0	R/(W)*	

Bit	Bit Name	Initial Value	R/W	Description
3	CMFEn3	0	R/(W)*	Cycle Match Flag Enm These bits cannot be set to 1 by software. Even if these bits are 1, meaning that the flag has not been cleared, the next cycle match can be input. In this case, 1 is rewritten to these bits. An interrupt request can be issued when the cycle match interrupt enable E bit (CMEEnm) in timer interrupt enable register En (TIEREn) is set to 1. To clear these bits, write 0 to these bits after reading them as 1. Flags for channel 0 in each subblock (CMFEn0) is automatically cleared by the ACK signal. 0: Cycle match has not occurred [Clearing conditions] • When writing 0 to these bits after reading them as 1 • When the ACK signal (status clearing) is asserted by the DMAC (supported only by channel 0) 1: Cycle match has occurred [Setting condition] • When counter E (TCNTEnm) is incremented while it is the same value as cycle setting register (CYLREnm)
2	CMFEn2	0	R/(W)*	
1	CMFEn1	0	R/(W)*	
0	CMFEn0	0	R/(W)*	

Notes: n = 0 to 6 (n denotes subblock number), m = 0 to 3 (m denotes channel number)

* Only 0 can be written to this bit after it is read as 1 to clear the flag. Writing 1 to this bit is ignored.

13.22.8 Timer Interrupt Enable Registers E0 to E6 (TIERE0 to TIERE6)

TIERE0 to TIERE6 are 8-bit readable/writable registers that enable and disable interrupt requests occurrence of a cycle match. The cycle match is a compare match between cycle setting register E (CYLRE) and timer counter E (TCNTE).

TIERE0 to TIERE6 can be read from and written to in byte or word units.

TIERE0 to TIERE6 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	CME En3	CME En2	CME En1	CME En0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

Note: n = 0 to 6 (correspond to subblocks E0 to E6)

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3 to 0	CMEEn3 to CMEEn0	All 0	R/W	Cycle Match Interrupt Enable Enm These bits enable and disable interrupt requests on cycle match of CYLREnm. When these bits are set to 1, interrupt requests can be issued by using cycle match flag Enm (CMFEnm) in timer status register En (TSREn). 0: CMFEnm interrupt requests are disabled 1: CMFEnm interrupt requests are enabled

Note: n = 0 to 6 (n denotes subblock number), m = 0 to 3 (m denotes channel number)

13.22.9 Timer Output Control Registers E0 to E6 (TOCRE0 to TOCRE6)

TOCRE0 to TOCRE6 are 8-bit readable/writable registers that select whether or not a signal on the PWM output pin (TOE) is inverted.

TOCRE0 to TOCRE6 can be read from and written to in byte or word units.

TOCRE0 to TOCRE6 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	TONE En3	TONE En2	TONE En1	TONE En0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

Note: n = 0 to 6 (correspond to subblocks E0 to E6)

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	TONEEn3	0	R/W	TOEnm Output Inversion Select These bits select whether or not a signal on the PWM output pin (TOE) is inverted. The output signal is inverted on the first edge of the Pϕ clock after duty modes are switched by the timer output control register. The operating state of the counter (TCNTEnm) has no effect on the mode switching. The initial level on the PWM output pin is low (TONEEnm = 0). 0: Signal is output as is on PWM output pin (TOEnm) 1: Inverted signal is output on PWM output pin (TOEnm)
2	TONEEn2	0	R/W	
1	TONEEn1	0	R/W	
0	TONEEn0	0	R/W	

Note: n = 0 to 6 (n denotes subblock number), m = 0 to 3 (m denotes channel number)

13.22.10 Timer Counters E00 to E63 (TCNTE00 to TCNTE63)

TCNTE00 to TCNTE63 are 16-bit readable/writable registers that are started by setting the TEE bit in the ATU-III master enable register (ATUENR), the subblock En start bit (STREn) in the timer start register E (TSTRE), and the counter Enm start bit (SSTREnm) in subblock starting register En (SSTREn).

The counter clock is selected by the TCNTEn clock select bits (CKSELEn[2:0]) in timer control register En (TCREn), and prescaler register En (PSCREn) of timer E.

These counters are initialized to H'0001 on cycle match with cycle setting register Enm (CYLREnm). For example, when the value in the cycle setting register is N and the counter value is to be incremented from N to N + 1, the counter value is changed to 1. This enables counting from 1 to N and PWM pulses with the cycle time of N is produced.

These counters can count from H'0001 to H'FFFF (when the cycle setting register value is H'FFFF).

When writing H'0000 to these counters, a PWM cycle is terminated and a new PWM cycle is started in the next clock cycle. While the counter value holds H'0000, the PWM output retains the previous value and outputs a level of 1 at the beginning of the new cycle. When the PWM cycle is terminated before duty cycle match, the duty cycle for that PWM cycle is 100% (1 is always output), that is, a level of 0 will not be output between PWM cycles. For details on writing H'0000 to these counters, see figure 13.33.

When TCNTEnm or CYLREnm is rewritten during the counter in operation, a cycle match may not occur even if the counter value reaches H'FFFF. In this case, the counter value is changed from H'FFFF to H'0000 in the next counter clock cycle. A PWM cycle is terminated in a way similar to writing H'0000. The counter value is incremented to H'0001 and a new PWM cycle is started. When the reload function is enabled, reloading of the cycle or duty cycle is also performed.

TCNTE00 to TCNTE63 can be read from and written to in word units.

TCNTE00 to TCNTE63 are initialized to H'0001 by a power-on reset or a transition to the hardware standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TCNTEnm[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Notes: 1. n = 0 to 6 (correspond to subblocks E0 to E6)
2. m = 0 to 3 (correspond to channels 0 to 3)

13.22.11 Cycle-Setting Registers E00 to E63 (CYLRE00 to CYLRE63)

CYLRE00 to CYLRE63 are 16-bit readable/writable registers that store the cycle of PWM. The settable value ranges from H'0001 to H'FFFF.

The value in CYLREnm is constantly compared with the value in the timer counter (TCNTEnm). When they match, the bit in the status register (TSRE) is set to 1 and TCNTEnm is initialized to H'0001. When the RLDENnm bit in the reload control register (RLDCREnm) is set to 1, the values in the cycle reload register (CRLDEnm) and duty cycle reload register (DRLDEnm) are transferred to cycle-setting register (CYLREnm) and duty cycle setting register (DTREnm).

To rewrite to CYLREnm during TCNTEnm in operation, note that the value to be set may lead to malfunction. When TCNTEnm in operation is rewritten, a cycle match may not be detected and TCNTEnm continues to be incremented even if the counter value exceeds the value in CYLREnm. In this case, unwanted PWM waveforms are output.

CYLRE00 to CYLRE63 can be read from and written to in word units.

CYLRE00 to CYLRE63 are initialized to H'FFFF by a power-on reset or a transition to the hardware standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CYLREnm[15:0]															
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Notes: 1. n = 0 to 6 (correspond to subblocks E0 to E6)
2. m = 0 to 3 (correspond to channels 0 to 3)

13.22.12 Duty Cycle Setting Registers E00 to E63 (DTRE00 to DTRE63)

DTRE00 to DTRE63 are 16-bit readable/writable registers that store the duty cycle of PWM. The settable value ranges from H'0000 to H'FFFF.

The value in DTREnm is constantly compared with the value in timer counter (TCNTEnm). When they match, the output level on the pin for the corresponding channel becomes low. When the values in CYLREnm and TCNTEnm match while the RLDENnm bit is set to 1, the value in DRLDEnm is reloaded to DTREnm.

The settable value in DTREnm ranges from 0 to the value in CYLREnm. When 0 is set, the duty cycle is 0% and the same value as CYLREnm is set, the duty cycle is 100%. DTREnm must be set to the value less than CYLREnm.

To rewrite to DTREnm during TCNTEnm in operation, note that the value to be set may lead to malfunction. When TCNTEnm in operation is rewritten, a duty cycle match may not be detected. In this case, unwanted PWM waveforms may be output.

DTRE00 to DTRE63 can be read from and written to in word units.

DTRE00 to DTRE63 are initialized to H'FFFF by a power-on reset or a transition to the hardware standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DTREnm[15:0]															
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Notes: 1. n = 0 to 6 (correspond to subblocks E0 to E6)
2. m = 0 to 3 (correspond to channels 0 to 3)

13.22.13 Cycle Reload Registers E00 to E63 (CRLDE00 to CRLDE63)

CRLDE00 to CRLDE63 are 16-bit readable/writable register that can be set to H'0001 to H'FFFF as the cycle of PWM outputs.

When the reload function is enabled, the value in this register is transferred to the cycle-setting register (CYLREnm) on cycle match.

CRLDE00 to CRLDE63 can be read from and written to in word units.

CRLDE00 to CRLDE63 are initialized to H'FFFF by a power-on reset or a transition to the hardware standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CRLDEnm[15:0]															
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Notes: 1. n = 0 to 6 (correspond to subblocks E0 to E6)
2. m = 0 to 3 (correspond to channels 0 to 3)

13.22.14 Duty Cycle Reload Registers E00 to E63 (DRLDE00 to DRLDE63)

DRLDE00 to DRLDE63 is a 16-bit readable/writable register that can be set to H'0000 to H'FFFF as the duty cycle.

When the reload function is enabled, the value in this register is transferred to the duty cycle-setting register (DTREnm) on cycle match.

DRLDE00 to DRLDE63 can be read from and written to in word units.

DRLDE00 to DRLDE63 are initialized to H'FFFF by a power-on reset or a transition to the hardware standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DRLDEnm[15:0]															
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Notes: 1. n = 0 to 6 (correspond to subblocks E0 to E6)
2. m = 0 to 3 (correspond to channels 0 to 3)

13.23 Operations of Timer E

Timer E consists of timer counter Enm (TCNTEnm), cycle-setting register Enm (CYLREnm), duty cycle setting register Enm (DTREnm), cycle reload register Enm (CRLDEnm), and duty cycle reload register Enm (DRLDEnm). Timer E can be used as a PWM timer.

TCNTEnm starts counting up when a channel is selected by subblock starting register En (SSTREn) after a subblock is selected by timer start register E (TSTRE). A logical zero level is output on pin TOEnm on the first edge of the counter clock after TCNTEnm matches duty cycle setting register Enm (DTREnm), or a logical one level is output on pin TOEnm on the first edge of the counter clock after TCNTEnm matches cycle setting register Enm (CYLREnm). After a match with the cycle setting register, the counter is set to H'0001 on the next counter clock edge and starts counting up again.

Subsequently duty-cycle and cycle match are repeated, producing a PWM output on pin TOEnm.

However, externally output level retains an initial value of 0 for one cycle which is from starting up the counter to the first cycle match.

The settable PWM cycle ranges from H'0001 to H'FFFF. The settable duty cycle ranges from 0% to 100%. When the duty cycle setting register is set to H'0000, the output level is 0 and remains unchanged (duty cycle = 0%). When the values in duty cycle setting register and cycle setting register are the same, the output level is 1 and remains unchanged (duty cycle = 100%). The value in duty cycle setting register must be equal to or less than the value in cycle setting register.

Figure 13.32 shows an operation example of PWM timer outputs for channel 0 in subblock E0. In this example, the duty cycle is changed every PWM cycle in the order of 75%, 67%, 0%, and 100%.

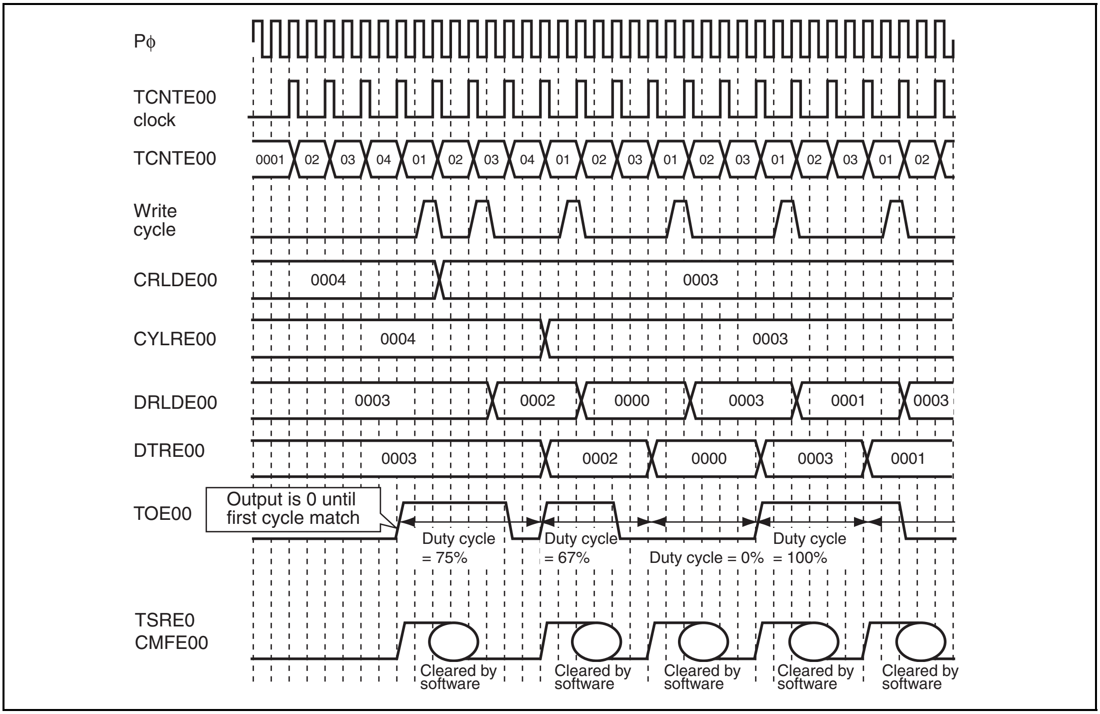


Figure 13.32 Operation of PWM (1)

The duty cycle setting and cycle setting registers have respective reload registers. When the values in the up-counter and cycle-setting register match, the values in the duty cycle reload and cycle reload registers are loaded to the duty cycle setting and cycle setting registers. The loaded data is updated in the next PWM cycle after loading. The reload function is enabled and disabled by the reload enable bit (RLDCREn) in the reload control register (RLDENEnm).

In timer E, a PWM output cycle is terminated by writing H'0000 to the counter (TCNTEnm). The counter value is changed from H'0000 to H'0001 on the next counter clock and the counter is restarted. When the counter value is changed, the values in the duty cycle reload and cycle reload registers are loaded to the duty cycle setting and cycle setting registers.

Figure 13.33 shows the PWM output cycle terminated by writing H'0000 to the counter and the counter restarted. The counter value is cleared to H'0000 by the writing. The output waveform (TOE00) is not changed. When the reload function is enabled after the writing, the values in the duty cycle reload and cycle reload registers are loaded to the duty cycle setting and cycle setting registers. At the same time, counting is restarted and PWM output is also restarted.

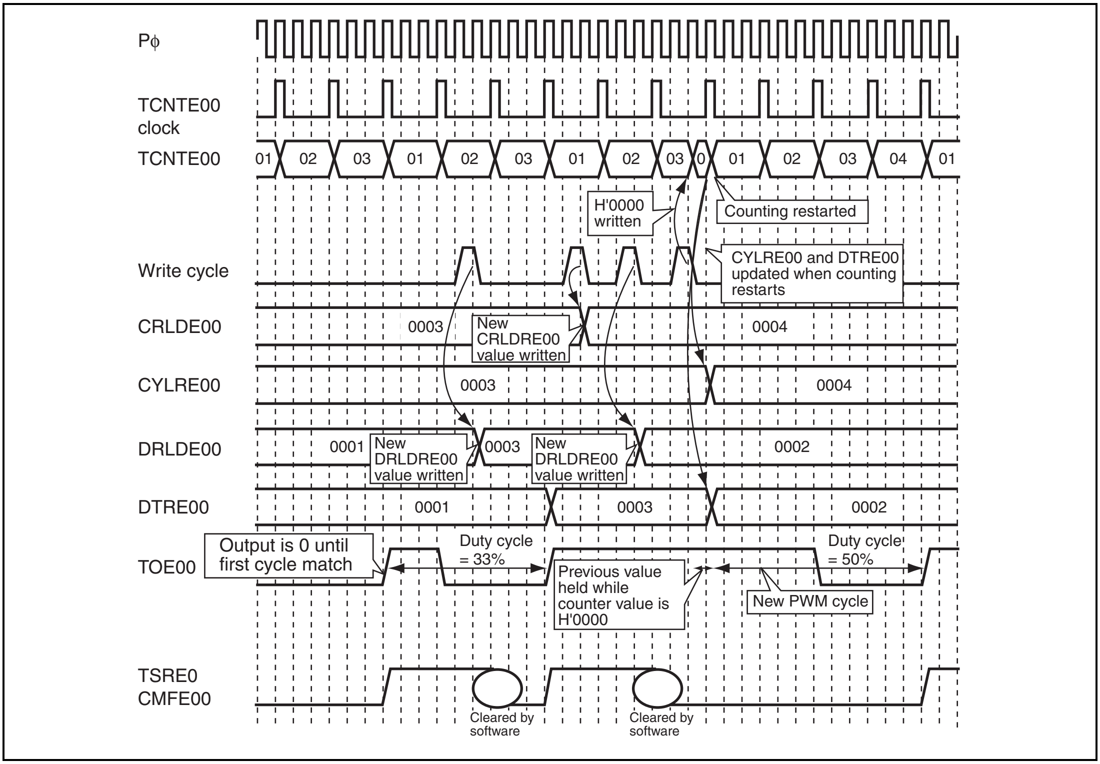


Figure 13.33 Operation of PWM (2)

A waveform is output in off-state duty (active-low output) mode by selecting the off-state duty mode in timer output control register En (TOCREn). The output waveform on pin TOEnm is inverted on the next Pϕ clock cycle after setting.

Figure 13.34 shows an example of a waveform when switching on- and off-state duty modes. By selecting the off-state duty mode before the counter is started, the initial output level on the PWM output pin TOE00 is 1. After the counter started until the first cycle match, the level on pin TOE00 retains 1. On the following cycle match and duty cycle match, the output levels are alternated. When the PWM cycle is forcibly terminated by writing H'0000 to the counter, TOE00 retains the previous value. At the timing in which the counter is incremented to H'0001, a new PWM cycle is started.

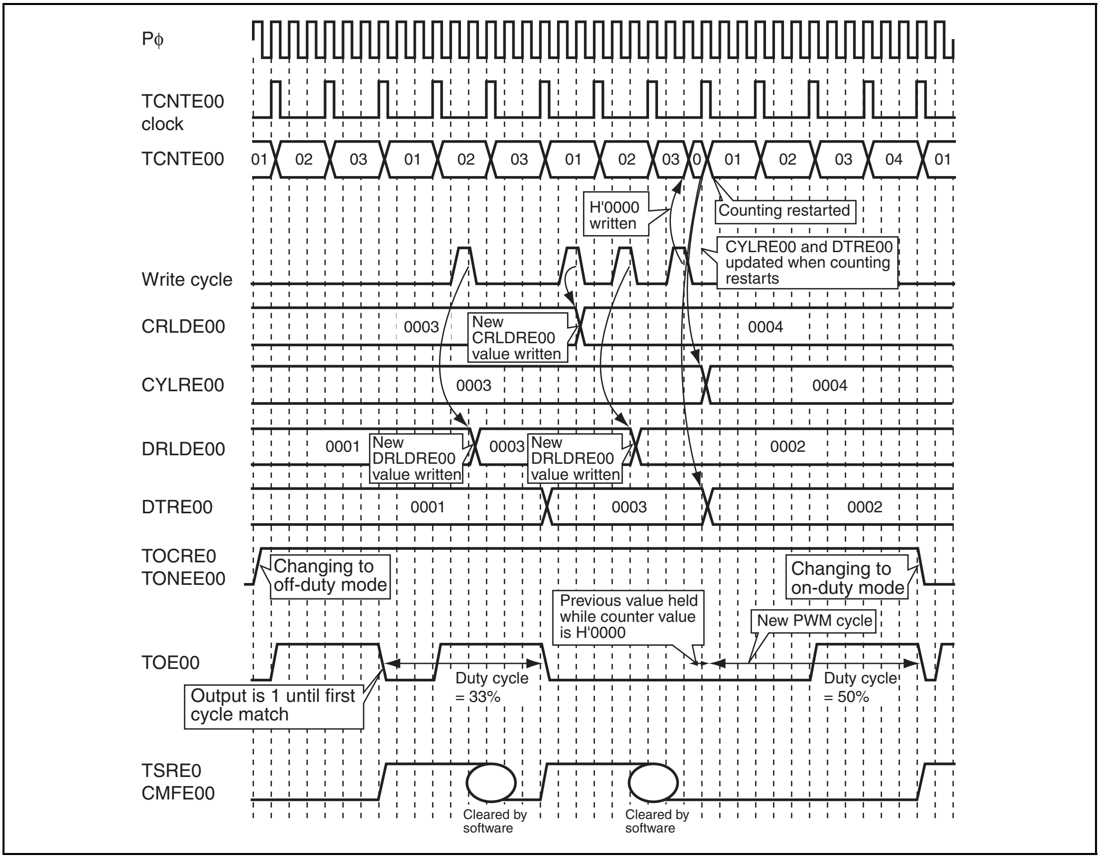


Figure 13.34 Operation of PWM (3)

13.24 Overview of Timer F

The timer F consists of 28 subblocks, featuring functions shown below.

- Edge counting in a specified period

  Counts the number of edges input to the external input pin (TIFnA)
- Valid edge interval counting

  Measures time until a specified number of edges is input to the external pin (TIFnA).
- Measurement of time during high/low input levels

  Measures a toal amount of time when a high or low level is input to the external input pin (TIFnA). The duration of measurement is designated as the number of pulses input to the external pin.
- Measurement of PWM input waveform timing

  Measures the off-duty period and cycle time of the PWM waveform input to the external pin (TIFnA). The duration of measurement is designated as the number of PWM cycles input to the external pin.
- Rotation speed/pulse measurement (for the subblock 12 to 15 only)

  Every time an edge is input to the external pin (TIFnA), the following values are retained — edge count, time stamp at edge input, edge input interval (cycle), and high/low input level immediately before input.
- Up/down event count (for the subblock 0 to 2 only)

  TIFnA of the two external pins (TIFnA, TIFnB) is used to count as the count source. TIFnB switches between upcounting and downcounting.
- Four-time multiplication event count (for the subblock 0 to 2 only)

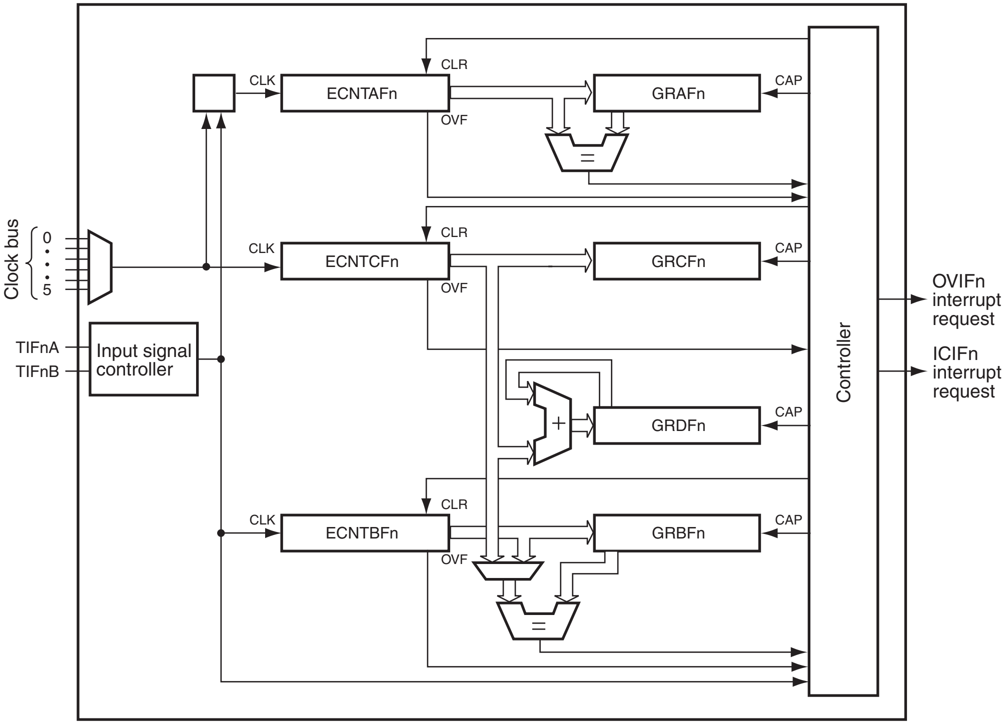
  Counting operation is executed using two external input pins (TIFnA, TIFnB) as the count sources. Signals in the pins switch between upcounting and downcounting.

Input signals from the external input pins TIFnA and TIFnB can be subject to the noise cancellation function using the input cancellation function.

13.24.1 Block Diagram

The timer F consists of 28 subblocks. Each subblock consists of such units as two 24-bit time counters (ECNTAFn, ECNTCFn), three 24-bit general registers (GRAFn, GRCFn, GRDFn), 16-bit event counter (ENCTBFn), 16-bit general register (GRBFn), input processing unit (edge detection, noise canceller), controllers etc.

Figure 13.35 is a block diagram of timer F.



[Legend]

ECNTAF:	Time counter AF	GRBF:	General-purpose register BF
ECNTBF:	Event counter F	GRCF:	General-purpose register CF
ECNTCF:	Time counter CF	GRDF:	General-purpose register DF
GRAF:	General-purpose register AF		

Notes: 1. One in TIFnB is provided for each of subblocks F0 to F2 and is used in the up/down counting and 4-times event counting modes. In other modes, the signal input on TIFnB does not affect the operation of timer Fn.

2. One register in GRDFn is provided for each of subblocks F12 to F15 and is used in the revolution rate/pulse counting mode.

3. n = 0 to 27

4. For subblocks 3 to 27, the TIFnA pins instead become TIF3 to TIF27, respectively.

Figure 13.35 Block Diagram of Subblocks of Timer F

13.24.2 Interrupts

The timer F can output two types of interrupts totaling 56 interrupts.

- OVIF0 to OVIF27 interrupts

  An interrupt is output when one of the three counters (ECNTAFn, ECNTBFn, ECNTCFn) in the subblock Fn has overflown or underflown (only in ECNTBFn). To which counter the interrupt belongs can be known by referring to the timer status register F (ISRF). This request is received by the INTC module and the designated processing is performed.
- ICIF0 to ICIF27 interrupts

  The interrupt is output when a count value capturing in the subblock Fn occurs. This request is received by the A-DMAC or INTC module. DMA transfer by A-DMAC enables to transfer captured data obtained by using compare match as a trigger to the on-chip SRAM or perform designated processing by interrupts. For details on DMA transfer by A-DMAC, see section 12, Automotive Direct Memory Access Controller (A-DMAC).

13.25 Description of Timer F Registers

13.25.1 Timer Start Register F (TSTRF)

TSTRF is a 32-bit readable/writable register that specifies whether to operate or stop each subblock (timer F0 to F27) in the timer F. Count operation is not executed unless TFE bit in ATU-III master enable register (ATUENR) is enabled even if the start bit in timer F is set to enable the count operation.

TSTRF can be read from and written to in byte, word, or longword units. However, the execution of access to the register as a longword unit is divided into two operations, i.e. reading or writing the respective words. Accordingly, the bits of both the higher and lower-order words cannot be accessed in the same clock cycle.

TSTRF is initialized to H'00000000 by a power-on reset or a transition to the hardware standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	STRF 27	STRF 26	STRF 25	STRF 24	STRF 23	STRF 22	STRF 21	STRF 20	STRF 19	STRF 18	STRF 17	STRF 16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	STRF 15	STRF 14	STRF 13	STRF 12	STRF 11	STRF 10	STRF 9	STRF 8	STRF 7	STRF 6	STRF 5	STRF 4	STRF 3	STRF 2	STRF 1	STRF 0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
27 to 0	STRF27 to STRF0	All 0	R/W	Counter Fn Start These bits specify whether to operate or stop two time counters in subblocks (ECNTAFn, ECNTCFn) and event counter (ECNTBFn). Counter value is retained at stop state. When this bit is set to 1 once again, the operation starts at the retained value. Count operation is not executed unless TFE bit in ATU-III master enable register (ATUENR) is enabled even if the start bit in timer F is set enable the count operation. 0: Stop the counting operation of ECNTAFn, ECNTBFn, and ECNTCFn. 1: Enable the counting operation of ECNTAFn, ECNTBFn, and ECNTCFn. Note: The prescaler is operating regardless of the setting of the counter F start bit, and not initialized at the start of counter. Therefore, during the time between the activation and the start of actual count operation by the above counter, hardware-related uncertainty shorter than the period of selected count source (resolution) accompanies

Note: n = 0 to 27 (correspond to subblocks F0 to F27)

13.25.2 Noise cancellation mode Channel Register 1F (NCMCR1F)

NCMCR1F is a 32-bit readable/writable register that selects the mode of operation of the noise canceller for each channel unit.

In the premature-transition cancellation mode, after a change in the level of the input signal has been detected, further changes in the level of the input signal that occur within a specified period are ignored. That is, changes with a specified period after each initial level change are regarded as noise in this mode.

The period is set by noise canceler registers in each of the applicable blocks and is counted by a noise canceler counter.

Figures 13.1 and 13.2 show the operation of the premature-transition cancellation and minimum time-at-level cancellation, using the TIA00 input signal of timer A as an example.

The edge for counting is detected from signals after noise removal in timers A, C, F, and J. Rising edges are being detected in figures 13.1 and 13.2.

NCMCR1F can be read from and written to in byte units.

This register is readable/writable only in byte or word units.

Note: The setting for this register is only effective when NCMF bit in the noise cancellation mode register (NCMR) of the common controller is 0.

Table 13.16 shows the truth table for settings that govern the noise cancellation mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	NCM1 F27	NCM1 F26	NCM1 F25	NCM1 F24	NCM1 F23	NCM1 F22	NCM1 F21	NCM1 F20	NCM1 F19	NCM1 F18	NCM1 F17	NCM1 F16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NCM1 F15	NCM1 F14	NCM1 F13	NCM1 F12	NCM1 F11	NCM1 F10	NCM1 F9	NCM1 F8	NCM1 F7	NCM1 F6	NCM1 F5	NCM1 F4	NCM1 F3	NCM1 F2	NCM1 F1	NCM1 F0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
27 to 0	NCM1F27 to NCM1F0	All 0	R/W	Channel n noise cancellation mode Specify the operation mode of noise canceller in channel n. 0: Subsequent edge cancellation mode 1: Preceding edge cancellation mode

Note: n = 0 to 27 (correspond to subblocks F0 to F27)

13.25.3 Noise Canceller Control Register F (NCCRF)

NCCRF is a 32-bit readable/writable register that specifies to enable/disable the noise canceller in the subblocks F0 to F27.

NCCRF can be read from and written to in byte, word, or longword units. However, the execution of access to the register as a longword unit is divided into two operations, i.e. reading or writing the respective words. Accordingly, the bits of both the higher and lower-order words cannot be accessed in the same clock cycle.

NCCRF is initialized to H'0000 0000 by a power-on reset or a transition to the hardware standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	NCEF 27	NCEF 26	NCEF 25	NCEF 24	NCEF 23	NCEF 22	NCEF 21	NCEF 20	NCEF 19	NCEF 18	NCEF 17	NCEF 16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NCEF 15	NCEF 14	NCEF 13	NCEF 12	NCEF 11	NCEF 10	NCEF 9	NCEF 8	NCEF 7	NCEF 6	NCEF 5	NCEF 4	NCEF 3	NCEF 2	NCEF 1	NCEF 0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
27 to 0	NCEF27 to NCEF0	All 0	R/W	Noise Canceller Enable Fn Specify to enable/disable the noise canceller in each subblock. Regarding the subblocks F2 to F0, each subblock has noise cancellers TIFnA and TIFnB but enabling/disabling these cancellers cannot be specified independently. Setting the NCEFn bit to 1 enables each noise canceller in TIFnA and TIFnB. When the noise canceller function is enabled and the level change in the external input signals (TIFnA, TIFnB) is detected, either subsequent edge cancel mode or preceding edge cancel mode starts according to the setting of noise cancellation mode register (NCMR) in common controller and noise cancellation mode channel register 1F (NCMCR1F) of timer F.

Bit	Bit Name	Initial Value	R/W	Description
27 to 0	NCEF27 to NCEF0	All 0	R/W	In subsequent edge cancel mode, when the input signal level change is detected, the change is output as the signal that has passed through noise canceling. Simultaneously, corresponding noise canceler counters (NCNTFA27 to NCNTFA0 and NCNTFB2 to NCNTFB0) start upcounting. The input signal level change is masked until a compare match occurs between the value in the noise canceler counter and the values in the noise cancel register (NCRFA27 to NCRFA0 and NCRFB2 to NCRFB0). When a compare match occurs, the input signal level at this moment is output as the signal after noise canceling. When these bits are cleared to 0 while NCNTFAn and NCNTFBn are in count operation, the count operation continues until a compare match occurs and the level change of the corresponding external input (TIFAn, TIFBn) is kept being masked. In preceding edge cancel mode, when the level change of the input signal is detected, the corresponding noise canceler counter (NCNTFA27 to NCNTFA 0 and NCNTFB2 to NCNTFB0) starts upcounting. If a level change of input signal is not detected during the period until a compare match occurs between the value in the noise canceler counter and the values in the noise cancel register (NCRFA27 to NCRFA0 and NCRFB2 to NCRFB0), the level change at the compare match is output as the signal after a noise cancellation. If a noise change is detected, this change is regarded as noise and the noise canceller does not change the signal after a noise cancel regarding that no level change of input signal occurred. When these bits are cleared to 0 while NCNTFAn and NCNTFBn are in count operation, the count operation continues to keep noise canceling processing until a compare match or input signal change occurs. For a operating example in cancel mode, see figures 13.1 and 13.2. 0: Noise cancel function of TIFnA and TIFnB is disabled. 1: Noise cancel function of TIFnA and TIFnB is enabled.

Note: n = 0 to 27 (correspond to subblocks F0 to F27)

13.25.4 Timer Control Registers F0 to F27 (TCRF0 to TCRF27)

TCRF0 to TCRF27 are 8-bit readable/writable registers that specify the operation mode of the subblocks F0 to F27.

TCRF0 to TCRF27 can be read from and written to in byte units.

TCRF0 to TCRF27 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	CKSELFn[2:0]			MDFn[2:0]			EGSELFn[1:0]	
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: n = 0 to 27 (correspond to subblocks F0 to F27)

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	CKSELFn [2:0]	000	R/W	Clock Select Fn Specify the clock sources for the two time counters (ECNTAFn, ECNTCFn) in the subblocks F0 to F27. Do not specify B'110, B'111. If specified, the operation is not guaranteed. 000: Clock bus 0 001: Clock bus 1 010: Clock bus 2 011: Clock bus 3 100: Clock bus 4 101: Clock bus 5 110: Reserved 111: Reserved

Bit	Bit Name	Initial Value	R/W	Description
4 to 2	MDFn[2:0]	000	R/W	Timer Operation Mode Fn Specify the operation mode for the corresponding subblocks F0 to F27. There are seven modes: up/down event count, four-time multiplication event count, edge counting in a specified period, valid edge interval counting, measurement of time during high/low input levels, measurement of PWM input waveform timing, and rotation speed/pulse measurement. 000: Edge counting in a specified period 001: Valid edge interval counting 010: Measurement of time during high/low input levels 011: Reserved 100: Measurement of PWM input waveform timing 101: Rotation speed/pulse measurement 110: Up/down event count 111: Four-time multiplication event count Note: Do not set rotation speed/pulse measurement for subblocks other than 12 to 15.

Bit	Bit Name	Initial Value	R/W	Description
1, 0	EGSELFn [1:0]	00	R/W	Edge Select Fn Specify the edge sense modes for event input (TIFnA) in the subblocks F0 to F27. Edge detection is done for signals that have passed through the noise canceller. Therefore, edge detection is done to the external input (TIFnA, TIFnB) if the noise cancel function is disabled, and to signals after noise cancel if the noise cancel function is enabled. While 'measurement of time during high/low input levels' is specified, when this bit selects the falling edge, measurement of time during high level is specified. When this bit selects the rising edge, measurement of time during low level is specified. Do not select both edges. While 'measurement of PWM input waveform timing' and 'rotation speed/pulse measurement' are specified, when this bit selects the rising edge, the period between the two rising edges is regarded as the PWM cycle and the low-level period is regarded as the off-duty period. If the falling edge is selected, the period between the two falling edges is regarded as the PWM cycle and the high-level period is regarded as the off-duty period. Do not select both edges. When 'up/down event count' mode and 'four-time multiplication event count' mode are specified, be sure to designate both the rising and falling edges. If otherwise selected, the operation is not guaranteed. 00: Edge detection disabled 01: Rising edge 10: Falling edge 11: Both edges Note: TIFnB pin is available only when 'up/down event count' and 'four-time multiplication event count' are specified. TIFnB operates always detecting both the rising and falling edges. In other modes, TIFnB does not detect edges.

Note: n = 0 to 27 (correspond to subblocks F0 to F27)

13.25.5 Timer Interrupt Enable Registers F0 to F27 (TIERF0 to TIERF27)

TIERF0 to TIERF27 are 8-bit readable/writable registers that specify whether to enable or disable the interrupt corresponding to the timer status register F (TSRF).

TIERF0 to TIERF27 can be read from and written to in byte units.

TIERF0 to TIERF27 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	OVE CFn	OVE BFn	OVE AFn	ICEFn
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

Note: n = 0 to 27 (correspond to subblocks F0 to F27)

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	OVECFn	0	R/W	Overflow Interrupt Enable CFn* Specifies whether to enable or disable the interrupt by OVFCFn to the status corresponding to the overflow of the time counter CFn (ECNTCFn) (in 'measurement of PWM input waveform timing' mode) or a compare match between ECNTCFn and GRBFn (in 'rotation speed/pulse measurement' mode) 0: Interrupt by OVFCFn disabled 1: Interrupt by OVFCFn enabled
2	OVEBFn	0	R/W	Overflow Interrupt Enable BFn* Specifies whether to enable or disable the interrupt by OVBCFn to the status corresponding to the overflow/underflow of the event counter BFn (ECNTBFn). 0: Interrupt by OVFBFn disabled 1: Interrupt by OVFBFn enabled

Bit	Bit Name	Initial Value	R/W	Description
1	OVEAFn	0	R/W	Overflow Interrupt Enable AFn* Specifies whether to enable or disable the interrupt by OVFAFn to the status corresponding to the overflow of the time counter AFn (ECNTAFn). 0: Interrupt by OVFAFn disabled 1: Interrupt by OVFAFn enabled
0	ICEFn	0	R/W	Input Capture Interrupt Enable Fn Specifies whether to enable or disable the interrupt by OVFAFn to the status corresponding to the input capture detection in the subblock Fn. 0: Interrupt by ICFFn disabled 1: Interrupt by ICFFn enabled

Note: * The overflow of interrupt of the subblock Fn is requested as the logical sum of the interrupts OVFAFn, OVFBFn, and OVFCFn, By referring to TSRFn, which counter generated the interrupt by overflow or underflow can be known.

13.25.6 Timer Status Registers F0 to F27 (TSRF0 to TSRF27)

TSRF0 to TSRF27 are 8-bit readable/writable registers that indicate overflows in the time counters A and C, overflow or underflow in the event counter, and input capture occurrence.

These flags are interrupt sources and requests the CPU interrupts if the corresponding bits to the timer interrupt enable register F0 to F27 (TIERF0 to TIERF27) enable interrupts.

TSRF0 to TSRF27 can be read from and written to in byte units.

TSRF0 to TSRF27 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	OVF CFn	OVF BFn	OVF AFn	ICFFn
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: n = 0 to 27 (correspond to subblocks F0 to F27)
* Only 0 can be written to this bit after it is read as 1 to clear it. Writing 1 is ignored.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	OVFCFn	0	R/(W)*	Overflow/Compare Match Flag CFn The values in this flag show different states depending on the operation mode. In 'measurement of PWM input waveform timing', the flag shows the overflow of the time counter C (ECNTCFn). In 'rotation speed/pulse measurement' mode, the flag shows a compare match between ECNTCFn and GRBFn. This flag cannot be set to 1 by software. 0: No overflow in ECNTCFn [Clearing condition] • After reading OVFCFn = 1, 0 is written to OVFCFn. 1: ECNTCFn overflows [Setting conditions] • Measurement of PWM input waveform timing mode When ECNTCFn overflows (H'FFFF FF → H'0000 00) • Rotation speed/pulse measurement mode When values in ECNTCFn and GRBFn (with the value zero extended to lower 8 bits) match
2	OVFBFn	0	R/(W)*	Overflow Flag BFn By this bit, overflow or underflow of the event counter BFn (ECNTBFn) can be monitored. This flag cannot be set to 1 by software. 0: No overflow or underflow in ECNTBFn [Clearing condition] • After reading OVFBFn = 1, 0 is written to OVFBFn. 1: ECNTBFn overflows or underflows [Setting condition] • When ECNTBFn overflows (H'FFFF → H'0000) or underflows (H'0000 → H'FFFF)

Bit	Bit Name	Initial Value	R/W	Description
1	OVFAFn	0	R/(W)*	Overflow Flag AFn By this bit, overflow of the time counter AFn (ECNTAFn) can be monitored. This flag cannot be set to 1 by software. 0: No overflow in ECNTAFn [Clearing condition] • After reading OVFAFn = 1, 0 is written to OVFAFn. 1: ECNTAFn overflows [Setting condition] • When ECNTAFn overflows (H'FFFF FF → H'0000 00)
0	ICFFn	0	R/(W)*	Input capture Flag Fn By this bit, the detection state of input capture in the subblock Fn can be monitored. This flag cannot be set to 1 by software. 0: Input capture is not detected in the subblock Fn. [Clearing conditions] • After reading ICFFn = 1, 0 is written to ICFFn. • When the capture output register (CDRF0 to CDRF27) are read by A-DMAC access. 1: Input capture in subblock Fn detected [Setting condition] • When input capture is detected in the subblock Fn.

Notes: n = 0 to 27 (correspond to subblocks F0 to F27)

* To clear the flag, only writing 0 after reading 1 is possible. Writing 1 is invalid.

13.25.7 Timer Counters AF0 to AF27 (ECNTAF0 to ECNTAF27)

ECNTAF0 to ECNTAF27 are 32-bit readable/writable registers.

This register, with one provided to each subblock, executes upcount operation using the input clock. One clock bus from clock buses 0 to 5 can be selected as the input clock according to the setting of the corresponding control register. The input clocks for ECNTAFn and ECNTCFn are the same. Clock source cannot be set independently.

When clearing the counter is done at the countup timing, ECNTAFn is cleared to H'00000100, and to H'00000000 in other cases.

ECNTAF0 to ECNTAF27 can be read from and written to in longword units.

ECNTAF0 to ECNTAF27 are initialized to H'0000 0000 by a power-on reset or a transition to the hardware standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECNTAFn[23:8]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECNTAFn[7:0]								-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Note: n = 0 to 27 (correspond to subblocks F0 to F27)

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	ECNTAFn [23:0]	All 0	R/W	Time Count AFn Upcounter A
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

13.25.8 Event Counters F0 to F27 (ECNTBF0 to ECNTBF27)

ECNTBF0 to ECNTBF27 are 16-bit readable/writable registers. This register, with one provided to each subblock, executes upcount/downcount operation using the input clock. The input clock is given two external input pins (TIFnA, TIFnB). The external pin and edge used to count differs according to the setting of the corresponding control register (operation mode and edge select). The input clock in each mode is listed in table 13.18.

When clearing the counter is done at the count-up timing, ECNTBFn is cleared to H'0001, and to H'0000 in other cases.

ECNTBF0 to ECNTBF27 can be read from and written to in word units.

ECNTBF0 to ECNTBF27 are initialized to H'0000 by a power-on reset or a transition to the hardware standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECNTBFn[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: n = 0 to 27(correspond to subblocks F0 to F27)

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	ECNTBFn [15:0]	All 0	R/W	Event Count Fn Up/down event counter

Table 13.18 Event Counter Input Clock and Count Edge for Each Timer F Operation Mode

Operation Mode	Input Clock	Count Edge
Edge counting in a specified period	TIFnA	Selectable by EGSELFn
Valid edge interval counting	TIFnA	Selectable by EGSELFn
Measurement of time during high/low input levels	TIFnA	Selectable by EGSELFn (other than both edges)
Measurement of PWM input waveform timing	TIFnA	Selectable by EGSELFn (other than both edges)
Rotation speed/pulse measurement	TIFnA	Selectable by EGSELFn (other than both edges)
Up/down event count	TIFnA (Count direction is specified by TIFnB level)	Both rising/falling edges
Four-time multiplication event count	TIFnA, TIFnB	Both rising/falling edges

13.25.9 Time Counters CF0 to CF27 (ECNTCF0 to ECNTCF27)

ECNTCF0 to ECNTCF27 are 32-bit readable/writable registers. This register, with one provided to each subblock, is enabled only in 'measurement of PWM input waveform timing' and 'rotation speed/pulse measurement' modes. This register does not execute count operation in other modes. This register executes upcount operation using the input clock. One clock bus from clock buses 0 to 5 can be selected as the input clock according to the setting of the corresponding control register. The input clocks for ECNTAFn and ECNTCFn are the same. Clock source cannot be set independently.

When clearing the counter is done at the external input timing or triggered by ECNTBFn, ECNTCFn is cleared synchronized with ECNTCFn count clock. At this moment, ECNTCFn is cleared to H'00000100.

ECNTCF0 to ECNTCF27 can be read from and written to in longword units.

ECNTAF0 to ECNTAF27 are initialized to H'0000 0000 by a power-on reset or a transition to the hardware standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECNTCFn[23:8]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECNTCFn[7:0]								-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Note: n = 0 to 27 (correspond to subblocks F0 to F27)

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	ECNTCFn [23:0]	All 0	R/W	Time Count CFn Upcounter C
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

13.25.10 General Registers AF0 to AF27 (GRAF0 to GRAF27)

GRAF0 to GRAF27 are 32-bit readable/writable registers. This register, with one provided to each subblock, has two functions such as input capture register and output compare register for the time counter Afn (ECNTAFn).

Do not set GRAFn to H'00000000 to function this register as the compare match register. Note that if H'00000000 is set, incorrect measurement may occur.

GRAF0 to GRAF27 can be read from and written to in longword units.

GRAF0 to GRAF27 are initialized to H'FFFF FF00 by a power-on reset or a transition to the hardware standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GRAFn[23:8]															
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GRAFn[7:0]								-	-	-	-	-	-	-	-
Initial value:	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Note: n = 0 to 27 (correspond to subblocks F0 to F27)

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	GRAFn [23:0]	All 1	R/W	General Registers AFn Input capture value or output compare match value for the time counter A.
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

13.25.11 General Registers BF0 to BF27 (GRBF0 to GRBF27)

GRBF0 to GRAB27 are 16-bit readable/writable registers. This register, with one provided to each subblock, has two functions such as input capture register and output compare register for the event counter (ECNTBFn).

Do not set GRBFn to H'0000 to function this register as the compare match register. Note that if H'0000 is set, incorrect measurement may occur.

GRBF0 to GRBF27 can be read from and written to in word units.

GRBF0 to GRBF27 are initialized to H'FFFF by a power-on reset or a transition to the hardware standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GRBFn[15:0]															
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: n = 0 to 27 (correspond to subblocks F0 to F27)

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	GRBFn [15:0]	All 1	R/W	General Registers BFn Input capture value or output compare match value for event counter.

13.25.12 General Registers CF0 to CF27 (GRCF0 to GRCF27)

GRCF0 to GRCB27 are 32-bit readable/writable registers. This register, with one provided to each subblock, has a function as the input capture register for the time counter (ECNTCFn). Triggered by a compare match between ECNTBFn and GRBn (in 'measurement of PWM input waveform timing' mode) or edge input of the TIFnA pin (in 'rotation speed/pulse measurement' mode), ECNTCFn count number is taken in at the next ECNTCFn upcount timing. These registers are valid only in 'measurement of PWM input waveform timing' or 'rotation speed/pulse measurement' mode. Capture operation is not executed in other modes.

GRCF0 to GRCB27 can be read from and written to in longword units.

GRCF0 to GRCB27 are initialized to H'FFFFFF00 by a power-on reset or a transition to the hardware standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GRCFn[23:8]															
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GRCFn[7:0]								-	-	-	-	-	-	-	-
Initial value:	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Note: n = 0 to 27 (correspond to subblocks F0 to F27)

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	GRCFn [23:0]	All 1	R/W	General Registers CFn Input capture value for the time counter C
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

13.25.13 General Registers DF12 to DF15 (GRDF12 to GRDF15)

GRDF12 to GRDF15 are 32-bit readable/writable registers. This register is provided to subblocks F12 to F15. Triggered by edge input of the TIFn pin, accumulated number in the time counter CFn (ECNTCFn) is taken in at the next ECNTAFn upcount timing. This register is valid only in 'rotation speed/pulse measurement' mode. Capture operation is not executed in other modes.

GRDF12 to GRDF15 can be read from and written to in longword units.

GRDF12 to GRDF15 are initialized to H'FFFFFF00 by a power-on reset or a transition to the hardware standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GRDFn[23:8]															
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GRDFn[7:0]								-	-	-	-	-	-	-	-
Initial value:	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Note: n = 12 to 15 (correspond to subblocks F12 to F15)

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	GRDFn [23:0]	All 1	R/W	General Registers DFn Input capture value for the time counter A
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

13.25.14 Capture Output Registers F0 to F27 (CDRF0 to CDRF27)

GRCF0 to GRCB27 are 32-bit readable/writable registers. This register is provided to each subblock. When this register is read, values in GRAFn, GRBFn, or ECNTBFN is read according to the operation mode. A 16-bit value in GRBFn is read from the upper 16 bits in CDRFn. In this case, the lower eight bits in CDRFn are read as 0.

CDRF0 to CDRF27 can be read from and written to in longword units.

CDRF0 to CDRF27 are initialized to H'FFFF0000 by a power-on reset or a transition to the hardware standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CDRFn[23:8]															
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CDRFn[7:0]								-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note: n = 0 to 27 (correspond to subblocks F0 to F27)

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	CDRFn [23:0]	H'FFFF00	R	Capture Output Register Fn Data stored in GRAFn or GRBFn is read according to the operation mode. Registers corresponding to various modes are listed below. Writing to these registers are ignored. Edge counting in a specified period mode: GRBFn Valid edge interval counting mode: GRAFn Measurement of time during high input levels mode: GRAFn Measurement of time during low input levels mode: GRAFn Measurement of PWM input waveform timing mode: GRAFn Rotation speed/pulse measurement mode: ECNTBFn Up/down event count mode mode: GRBFn Four-time multiplication event count mode: GRBFn

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

13.25.15 Noise Canceler Counters FA0 to FA27 (NCNTFA0 to NCNTFA27)

NCNTFA0 to NCNTFA27 are 8-bit readable/writable registers. When the noise canceler control register Fn (NCCRFn) enables the noise canceler function, these registers start upcount operation using the count clock for noise canceler supplied by the pre-scaler, triggered by the level change in the external input pin (TIFnA).

The timer F can output two types of interrupts totaling 40 interrupts.

Two types of operation modes — subsequent edge cancel mode and preceding edge cancel mode — can be set according to the setting of the timer F niose cancel mode bit (NCMF) in the noise cancelltion mode register (NCMR) of the common controller and noise cancellation mode bit (NCM1F0 to NCM1F27) in noise cancellation mode channel register 1F (NCMCR1F) of timer F.

- Subsequent edge cancel mode

  NCNTFAn starts upcount operation triggered by the level change of input signal in TIFnA under the condition that the NCEFn bit is set to 1 and NCNTFAn is not in count operation. When the count number matches the value in the noise cancel register FAn (NCRFAn), this register stops the count operation, clearing the count value to H'00 synchronizing with the next Pϕ clock. NCNTFAn executes the count operation regardless of the setting of the TFE bit in the ATU-III master enable register (ATUENR).

  A level change at the start of count operation is output as it is as the signal that passed through the noise canceling operation. Although this signal is the subject of edge detection, the signal does not change because any input level change is masked until the count number matches the value of NCRFAn.

  Even if the NCEFn bit is cleared during the count operation, the count operation continues until the count number matches the value of NCRFAn. The input signal is masked during all that time.

- Preceding edge cancel mode

  NCNTFAn starts upcount operation triggered by the level change of input signal in TIFnA under the condition that the NCEFn bit is set to 1 and NCNTFAn is not in count operation. When the level change of the input signal occurs during the count operation or the count number matches the value in the noise cancel register FAn (NCRFAn), this register stops the count operation, clearing the count value to H'00 synchronizing with the next Pϕ clock. NCNTFAn executes the count operation regardless of the setting of the TFE bit in the ATU-III master enable register (ATUENR).

  Signals that passed through the noise canceling operation can change only when the count number matches the value of NCRFAn according to the level change at the count strart. If the count operation stops before the count number matches the value of NCRFAn, signals that passed through the noise canceling operation do not change because the level changes at the start or stop of counting are masked.

  Even if the NCEFn bit is cleared during the count operation, the count operation and noise canceling processing continue until a compare match occurs or the level of the input signal changes.

NCNTFA0 to NCNTFA27 can be read from and written to in byte units.

NCNTFA0 to NCNTFA27 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	NCNTFAn[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: n = 0 to 27 (correspond to subblocks F0 to F27)

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	NCNTFAn [7:0]	All 0	R/W	Noise Cancel Count FAn 8-bit count value

13.25.16 Noise Canceler Counters FB0 to FB2 (NCNTFB0 to NCNTFB2)

NCNTFB0 to NCNTFB2 are 8-bit readable/writable registers.

These registers are available only in 'up/down event count' mode and '4-time multiplication event count' mode.

When the noise canceler control register Fn (NCCRFn) enables the noise canceler function, these registers start upcount operation using the count clock for noise canceler supplied by the pre-scaler, triggered by the level change in the external input pin (TIFnB).

Two types of operation modes — subsequent edge cancel mode and preceding edge cancel mode — can be set according to the setting of the timer F noise cancellation mode bit (NCMF) in the noise cancelltion mode register (NCMR) of the common controller and the noise cancellation mode bits (NCM1F0 to NCM1F27) in noise cancellation mode channel register 1F (NCMCR1F) of timer F.

- Subsequent edge cancel mode

  NCNTFBn starts upcount operation triggered by the level change of input signal in TIFnB under the condition that the NCEFn bit is set to 1 and NCNTFBn is not in count operation. When the count number matches the value in the noise cancel register FBn (NCRFBn), this register stops the count operation, clearing the count value to H'00 synchronizing with the next P clock. NCNTFBn executes the count operation regardless of the setting of the TFE bit in the ATU-III master enable register (ATUENR).

  A level change at the start of count operation is output as it is as the signal that passed through the noise canceling operation. Although this signal is the subject of edge detection, the signal does not change because any input level change is masked until the count number matches the value of NCRFBn.

  Even if the NCEFn bit is cleared during the count operation, the count operation continues until the count number matches the value of NCRFBn. The input signal is masked during all that time.

- Preceding edge cancel mode

  NCNTFBn starts upcount operation triggered by the level change of input signal in TIFnB under the condition that the NCEFn bit is set to 1 and NCNTFBn is not in count operation. When the level change of the input signal occurs during the count operation or the count number matches the value in the noise cancel register FBn (NCRFBn), this register stops the count operation, clearing the count value to H'00 synchronizing with the next Pϕ clock. NCNTFBn executes the count operation regardless of the setting of the TFE bit in the ATU-III master enable register (ATUENR).

Signals that passed through the noise canceling operation can change only when the count number matches the value of NCRFBn, according to the level change at the count strart. If the count operation stops before the count number matches the value of NCRFBn, signals that passed through the noise canceling operation do not change because the level changes at the start or stop of counting are masked.

Even if the NCEFn bit is cleared during the count operation, the count operation and noise canceling processing continue until a compare match occurs or the level of the input signal changes.

NCNTFB0 to NCNTFB27 can be read from and written to in byte units.

NCNTFB0 to NCNTFB27 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	NCNTFBn[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: n = 0 to 2 (correspond to subblocks F0 to F2)

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	NCNTFBn [7:0]	All 0	R/W	Noise Cancel Count FBn 8-bit count value

13.25.17 Noise Cancel Registers FA0 to FA27 (NCRFA0 to NCRFA27)

NCRFA0 to NCRFA27 are 8-bit readable/writable registers that set the upper limit of the noise canceler counter (NCNTFAn). Noise in the period up to 1.64 ms (when Pϕ = 20MHz) can be cancelled by setting the register to H'FF.

Two types of operation modes — subsequent edge cancel mode and preceding rdge cancel mode — can be set according to the setting of the timer F niose cancel mode bit (NCMF) in the noise cancellation mode register (NCMR) of the common controller and the noise cancellation mode bits (NCM1F0 to NCM1F27) in noise cancellation mode channel register 1F (NCMCR1F) of timer F.

- Subsequent edge cancel mode

  While NCNTFAn is in count operation, the level change of the subsequent input signal is masked. Values in NCNTFAn and NCRFAn are always compared. If a compare match occurs, these registers clear the value in NCNTFAn synchronizing with the next Pϕ clock, stop the count operation, and cancel the masking of the input signal.
- Preceding edge cancel mode

  While NCNTFAn is in count operation, noise canceler processing waiting state is entered. Values in NCNTFAn and NCRFAn are always compared. If a compare match occurs, these registers clear the value in NCNTFAn synchronizing with the next Pϕ clock, stop the count operation, and then cancel the masking of the input signal and the noise canceler outputs the input signal that has passed through the noise canceling processing.

NCRFA0 to NCRFA27 can be read from and written to in byte units.

NCRFA0 to NCRFA27 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	NCTFAn[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: n = 0 to 27 (correspond to subblocks F0 to F27)

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	NCTFAn [7:0]	All 0	R/W	Noise Cancel Time FAn TIFnA noise cancel period (8-bit compare value)

13.25.18 Noise Cancel Registers FB0 to FB2 (NCRFB0 to NCRFB2)

NCRFB0 to NCRFB2 are 8-bit readable/writable registers that set the upper limit of the noise canceler counter (NCNTFBn). Noise in the period up to 1.64 ms (when Pϕ = 20MHz) can be cancelled by setting the register to H'FF.

These registers are available only in 'up/down event count' mode and '4-time multiplication event count' mode.

Two types of operation modes — subsequent edge cancel mode and preceding rdge cancel mode — can be set according to the setting of the timer F niose cancel mode bit (NCMF) in the noise cancelltion mode register (NCMR) in the common controller and the noise cancellation mode bits (NCM1F0 to NCM1F27) in noise cancellation mode channel register 1F (NCMCR1F) of timer F.

- Subsequent edge cancel mode

  While NCNTFBn is in count operation, the level change of the subsequent input signal is masked. Values in NCNTFBn and NCRFBn are always compared. If a compare match occurs, these registers clear the value in NCNTFBn synchronizing with the next Pϕ clock, stop the count operation, and cancel the masking of the input signal.
- Preceding edge cancel mode

  While NCNTFBn is in count operation, noise canceler processing waiting state is entered. Values in NCNTFBn and NCRFBn are always compared. If a compare match occurs, these registers clear the value in NCNTFBn synchronizing with the next Pϕ clock, stop the count operation. Simultaneously the noise canceler outputs the input signal that has passed through noise canceling processing.

NCRFB0 to NCRFB2 can be read from and written to in byte units.

NCRFB0 to NCRFB2 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	NCTFBn[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: n = 0 to 2 (correspond to subblocks F0 to F2)

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	NCTFBn [7:0]	All 0	R/W	Noise Cancel Time FBn TIFnB noise cancel period (8-bit compare value)

13.26 Operations of Timer F

13.26.1 Edge Counting

When a period over which edges are counted is set in GRAFn, the number of edges within the period is obtained in GRBFn. When no edge is detected within the period, 0 is set to GRBFn. The period set to count is eqivqlent to the cycle of the ECNTAFn clock (GRAFn value). Operation of the timer Fn is described below. Figure 13.36 shows an operation example. In this example, eight edges are input to 12 cycles of the count source clock. Timer counter ECNTAFn and event counter ECNTBFn are driven by the ECNTAFn and ECNTBFn clocks, respectively.

Operation of registers in edge counting mode is shown below.

- ECNTAFn: Measures time using one of the clock buses 0 to 5. When a compare match is detected, the count value is cleared synchronized with the next Pϕ clock.
- ECNTBFn: Counts edges of the signals provided from TIFnA input. Edge types subject to count can be selected from among rising, falling, or both edges. The example given here counts the falling edges. A delay of two cycles in TIFnA occurs because of synchronization processing. When a compare match in ECNTAFn is detected, the count number is cleared synchronized with the next Pϕ clock. In a case where edges subject to count are given simultaneously at a count clearing by a compare match, both operations are regarded to be done in one cycle, setting the count value to H'0001. Figure 13.38 shows an example of this.
- GRAFn: Functions as the compare match register for ECNTAFn. A compare match is detected when the count values in ECNTAFn and GRAFn agree.
- GRBFn: Functions as the capture register for ECNTBFn. When a compare match in ECNTAFn is detected, this register captures the ECNTBFn count number synchronizing with the next Pϕ clock.
- ICFFn flag: After detecting a compare match in ECNTAFn, sets the ICFFn flag synchronized with the next Pϕ clock.
- ECNTCFn, GRCFn, GRDFn: Do not function

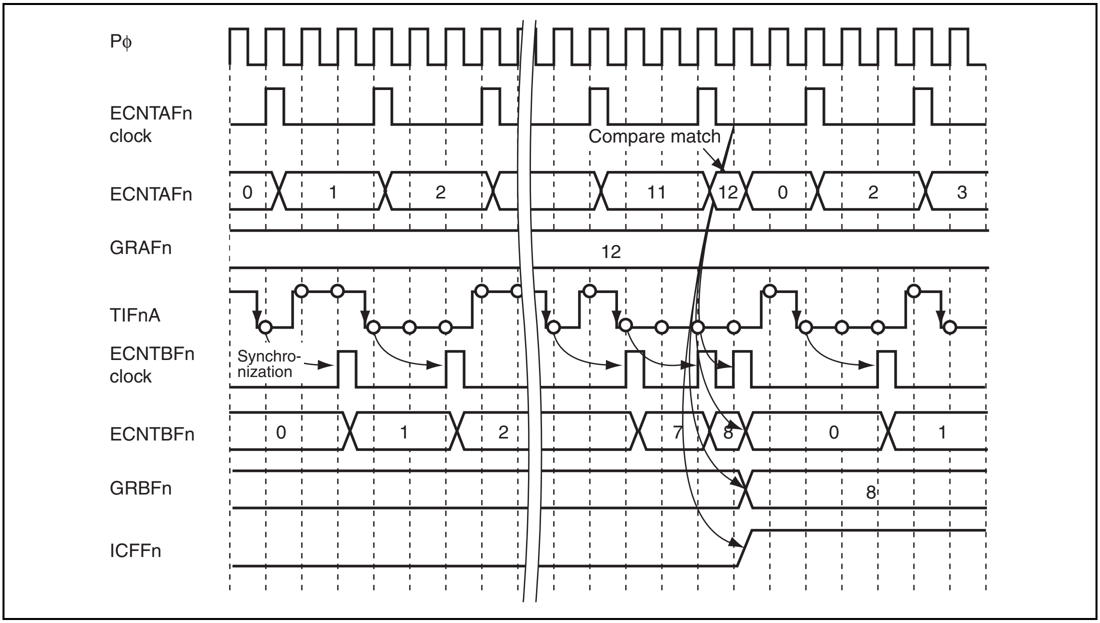


Figure 13.36 Operation Example of Edge Count in a Given Time

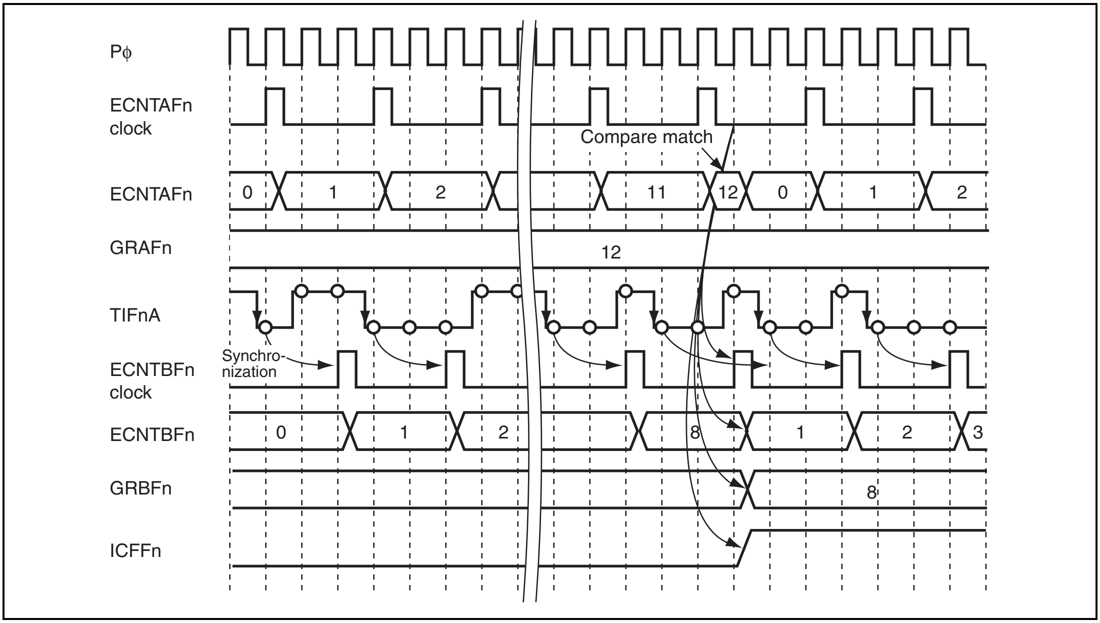


Figure 13.37 Operation Example of Edge Count in a Given Time (Compare Match and Event Occur Simultaneously)

13.26.2 Valid Edge Interval Counting

When a number of edges are set in GRBFn, the time necessary to count these edges is notified to GRAFn. The average of input edge intervals is obtained by dividing the time by the number of edges. The outcome is given as the unit of the ECNTAFn count source clock (GRAFn). Operation of the timer Fn is described below. Figure 13.38 shows an operation example. In this example, 13 cycles of the counter clock are needed to detect 12 input edges. Timer counter ECNTAFn and event counter ECNTBFn are driven by the ECNTAFn and ECNTBFn clocks, respectively.

Operation of registers in 'valid edge counting' mode is shown below.

- ECNTAFn: Measures time using one of the clock buses 0 to 5. When a compare match between ECNTBFn and GRBFn is detected, the count value is cleared synchronized with the next ECNTAFn clock. Since ECNTAFn count clear occurs at the same time with countup, the cleared value becomes H'00000100.
- ECNTBFn: Counts edges provided from TIFnA. Edge types subject to count can be selected from among rising, falling, or both edges. The example given here counts the falling edges. A delay of two cycles in TIFnA occurs because of synchronization processing. When a compare match is detected, the count number is cleared synchronized with the next Pϕ clock.
- GRAFn: Functions as the capture register for ECNTAFn. When a compare match in ECNTBFn is detected, this register captures the ECNTAFn count number synchronizing with the next ECNTAFn clock.
- GRBFn: Functions as the compare match register for ECNTBFn. A compare match is detected when the count values in ECNTBFn and GRBFn agree.
- ICFFn flag: After detecting a compare match in ECNTBFn, sets the ICFFn flag synchronized with the next ECNTAFn clock.
- ECNTCFn, GRCFn, GRDFn: Do not function.

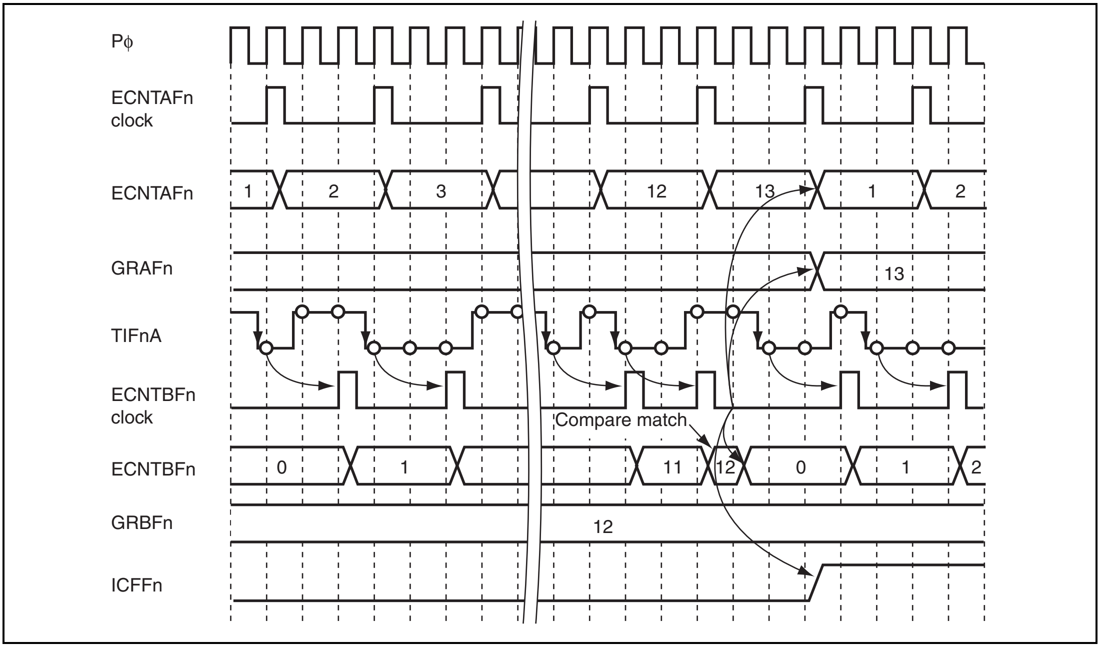


Figure 13.38 Operation Example of Valid Edge Interval Counting

13.26.3 Measurement of Time during High/Low Input Levels

Measures the time while TIFnA is driven high or low. The time obtained is indicated using the ECNTAFn clock source as the standard. The width of the measument time is specified to GRBFn in the form of the pulse number provided for TIFnA (GRBFn value). Operation of the timer F is described below. Figure 13.39 shows an operation example. This is the example in which the high level periods of the three pulses are measured as nine count source cycles. Timer counter ECNTAFn and event counter ECNTBFn are driven by the ECNTAFn and ECNTBFn clocks, respectively.

Operation of registers in 'counting high or low level of input' mode is shown below.

- ECNTAFn: Executes upcount using one of the clock buses 0 to 5 as a count source and TIFnA level as enable. Therefore, the time period in which TIFnA is in high level is measured. After detecting a compare match in ECNTBFn, this register clears the count number synchronizng with the next count source clock. If TIFnA is driven high at clearing count by the compare match, the count value becomes H'00000100. Figure 13.40 is an example of this.

- ECNTBFn: Counts the falling edge of TIFnA. A delay of two cycles occurs because of synchronization processing. When a compare match is detected, the count number is cleared synchronized with the next Pϕ clock.
- GRAFn: Functions as the capture register for ECNTAFn. When a compare match in ECNTBFn is detected, this register captures the ECNTAFn count number synchronizing with the next ECNTAFn clock.
- GRBFn: Functions as the compare match register for ECNTBFn. A compare match is detected when the count number in ECNTBFn and GRBFn agree.
- ICFFn flag: After detecting a compare match in ECNTBFn, sets the ICFFn flag synchronized with the next ECNTAFn clock.
- ECNTCFn, GRCFn, GRDFn: Do not function.

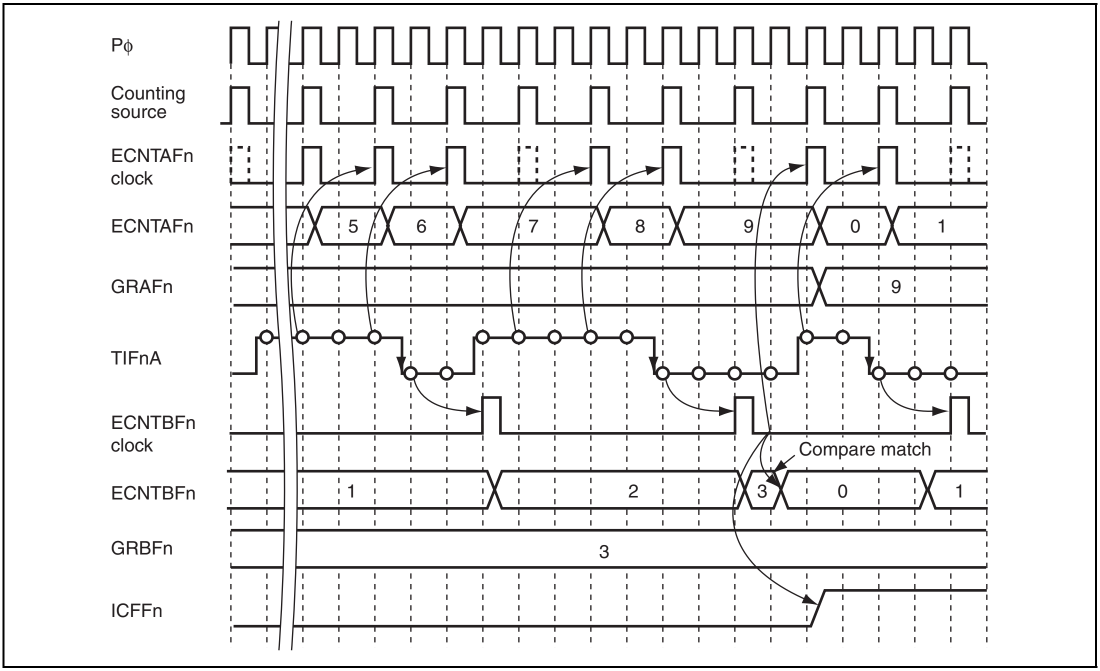


Figure 13.39 Operation Example of Measurement of Time during High Input Levels

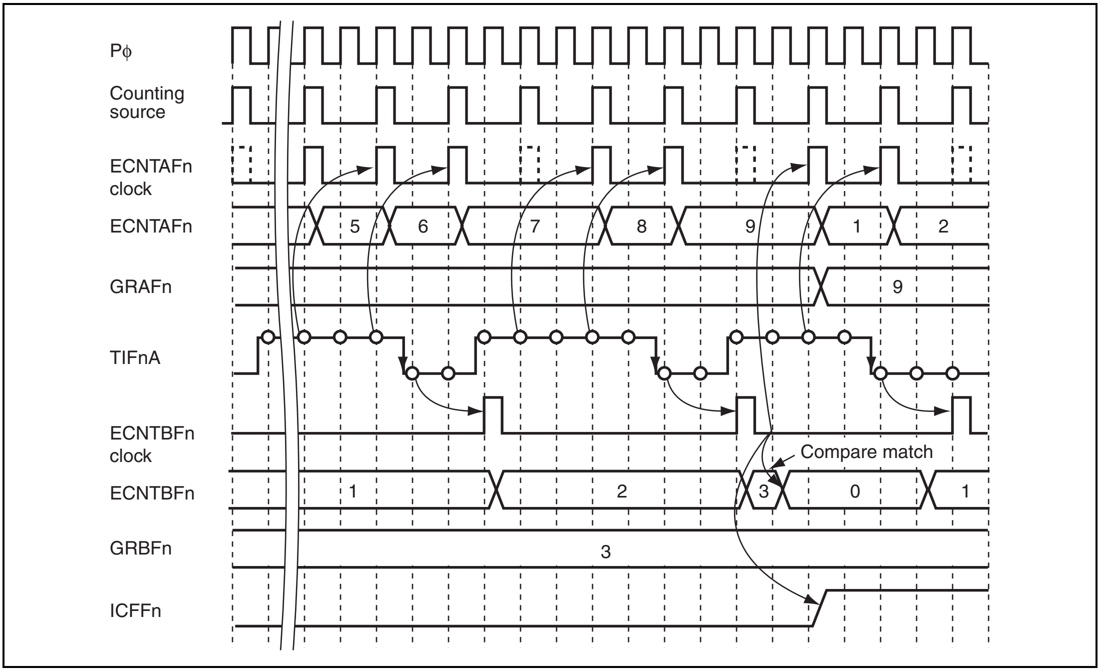


Figure 13.40 Operation Example of Measurement of Time during High Input Levels (TIFnA is in High Level When Capture is in Operation)

13.26.4 Measurement of PWM Input Waveform Timing

Measures the off-duty (non-active) period and cycle time of the PWM waveform input to TIFnA. The off-duty period is measured as the period of either the high or low level input on TIFnA, and the PWN cycle is measured as the interval between two rising or falling edges. Both are measured concurrently.

The measured time is expressed in the number of cycles of the clock source for ECNTAFn. The duration of the measument is set in GRBFn, which is specified as the number of PWM pulses input to TIFnA.

Operation of timer F is described below. Figure 13.41 shows an operation example. This is the example in which two PWM cycles in PWM waveform are measured as six counter clock cycles and the off-duty period (low-level period) is measured as four counter clock cycles.

The clocks for ECNTAFn, ECNTBFn, and ECNTCFn in this example provide the timing of counting or clearing operation of the time counter ECNTAFn, event counter ECNTBFn, and ECNTCFn, respectively.

The operation of each register in 'measurement of PWM input waveform timing' mode are as follows:

- ECNTAFn: Executes upcount using one of the clock buses 0 to 5 as a count source and TIFnA level as enable. Therefore, the time period in which TIFnA is in low level is measured. After detecting a compare match in ECNTBFn, this register clears the count number synchronizng with the next count source clock. If TIFnA is driven low at clearing count by the compare match, the count value becomes H'00000100.
- ECNTBFn: Counts the rising edge of TIFnA. A delay of two cycles occurs because of sybchronization processing. When a compare match is detected, the count number is cleared synchronized with the next Pϕ clock.
- ECNTBFn: Counts the falling edge of TIFnA. A delay of two cycles occurs because of synchronization processing. When a compare match is detected, the count number is cleared synchronized with the next Pϕ clock.
- GRAFn: Functions as the capture register for ECNTAFn. When a compare match in ECNTBFn is detected, this register captures the ECNTAFn count number synchronizing with the next ECNTAFn clock.
- GRBFn: Functions as the compare match register for ECNTBFn. A compare match is detected when the count number in ECNTBFn and GRBFn agree.
- ECNTCFn: Measures time using the same count source as ECNTAFn. This register clears the count number synchronizing with the next ECNTAFn clock after detecting a compare match in ECNTBFn. Since ECNTCFn count clear occurs at the same time eith ccountup, the cleared value is H'00000100.
- GRCFn: Functions as the capture register for ECNTCFn. This register captures the ECNTCFn count number synchronizing with the next ECNTAFn clock after detecting a compare match in ECNTBFn.
- ICFFn flag: After detecting a compare match in ECNTBFn, sets the ICFFn flag synchronized with the next ECNTAFn clock.
- GRDFn: Does not function.

Therefore, ECNTBFn (GRBFn) and ECNTAFn (GRAFn) are operating in measurement of time during low input levels mode and ECNTBFn (GRBFn) and ECNTCFn (GRCFn) are operating in valid edge interval counting mode.

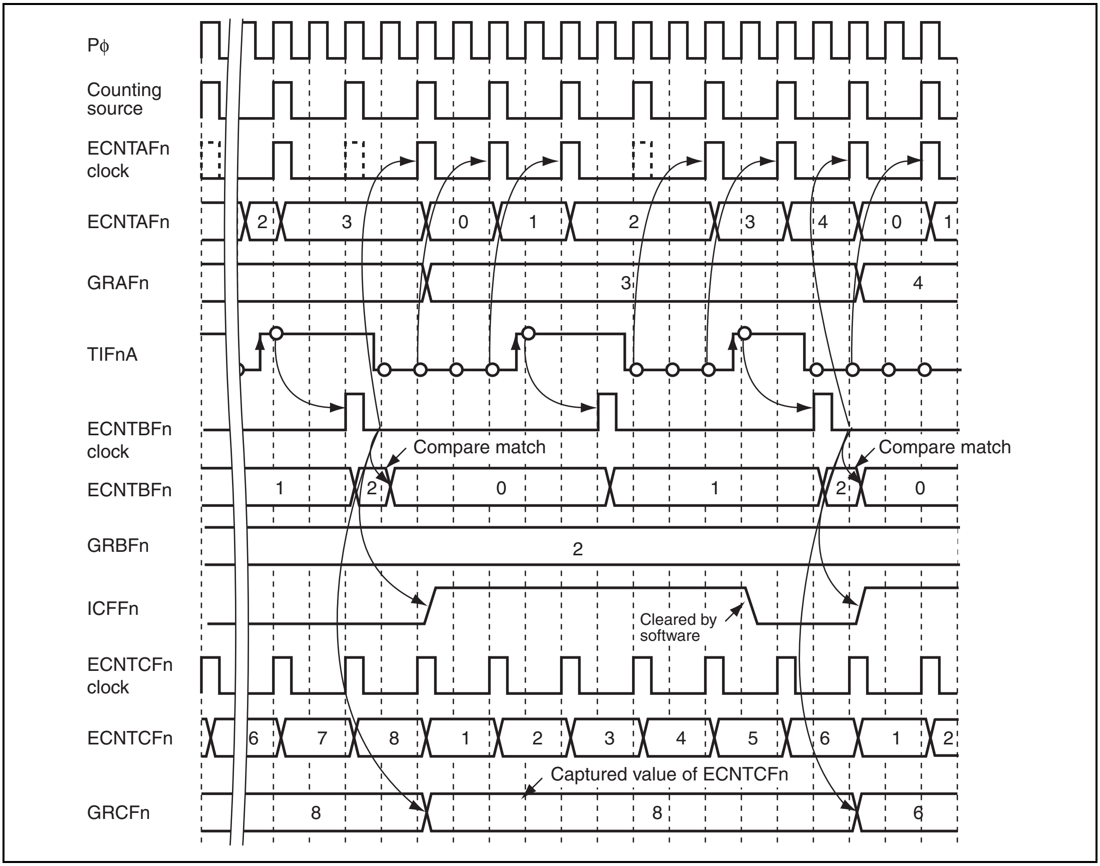


Figure 13.41 Operation Example of Measurement of PWM Input Waveform Timing

13.26.5 Rotation Speed/Pulse Measurement

Measures the number of edges input to TIFnA, the edge input time (time stamp), the off-duty period in the PWM waveform that emerges between the last input edge and the edge this time, and PWM cycle time.

The time obtained is expressed using the ECNTAFn clock source as the standard. The maximum interval of edge input can be set to GRBFn, which enables to output an interrupt request if the edge input interval exceeds the maximum value.

At this moment, the timer F operates as shown below. Figure 13.42 shows an example of operation. The ECNTAFn clock, ECNTBFn clock, and ECNTCFn clock show the timing of count operation or clearing for the time counter ECNTAFn, event counter ECNTBFn, and ECNTCFn, respectively.

The operations of each register in 'rotation speed/pulse measurement' mode are as follows:

- ECNTAFn: Executes upcount using one of the clock buses 0 to 5 as a count source and TIFnA input level as enable. Therefore, the time period in which TIFnA is in low level is measured. After inputting the edge to TIFnA, this register clears the count number synchronizng with the next count source clock. If TIFnA is driven low at clearing count, the count value becomes H'00000100.
- ECNTBFn: Counts the rising edge of TIFnA. A delay of two cycles occurs because of sybchronization processing.
- GRAFn: Functions as the capture register for ECNTAFn. This register captures the ECNTAFn count number synchronizing with the next ECNTAFn clock after inputting the edge to TIFnA.
- GRBFn: Functions as the capture register for ECNTCFn. When the ECNTCFn count and the value in lower eight bits in GRBFn extended with 0 match, this register detects a compare match and set the OVFCFn to 1.
- ECNTCFn: Measures time using the same count source as ECNTAFn. This register clears the count number synchronizing with the next ECNTAFn clock after inputting the edge to TIFnA. Since ECNTCFn count clear occurs in the same timing, the cleared value is H'00000100.
- GRCFn: Functions as the capture register for ECNTCFn. This register captures the ECNTCFn count number synchronizing with the next ECNTAFn clock after inputting the edge to TIFnA.
- GRDFn: Functions as the capture register for ECNTCFn. This register captures the ECNTAFn, whose number being accumulated to GRDFn, synchronizing with the next ECNTAFn clock after inputting the edge to TIFnA. The value to be added is the ECNTCFn value before clearing.
- ICFFn flag: Sets the ICFFn flag synchronizing with the next ECNTAFn clock after inputting the edge to TIFnA.
- OVFCFn flag: Sets the OVFCFn synchronizing with the next Pϕ clock after the values in ECNTCFn and GRBFn (in lower eight bits extended with 0) match.

While the ICFFn flag is set to 1, information on edge number, off duty cycle, PWM cycle, and edge input time can be obtained by reading ECNTBFn, GRAFn, GRCFn, and GRDFn, respectively. The capture timing of GRAFn, GRCFn, and GRDFn synchronizes with the count clock of ECNTAFn. Note that if the edge input cycle is shorter than the ECNTAFn count clock cycle, incorrect measurement may occur.

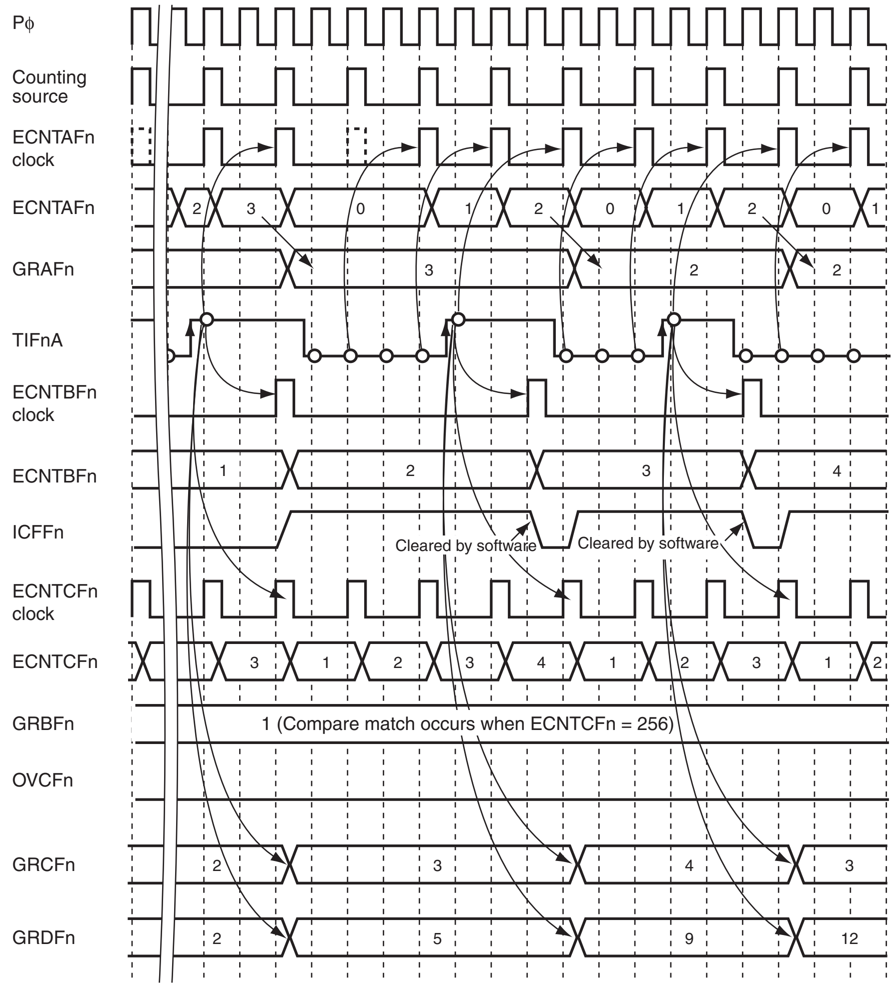


Figure 13.42 Operation Example of Rotation Speed/Pulse Measurement

13.26.6 Up/Down Event Count

This register uses the TIFnA pin, one of the two external input pins (TIFnA, TIFnB), as the count source, and TIFnB switches upcount to and from downcount. If a count period is designated to GRAFn, the count number after designation can be obtained in GRBFn. The counting period is the period of ECNTAFn count source clock (GRAFn value).

At this moment, the timer F operates as shown below. Figure 13.43 shows an example of operation. The ECNTAFn clock and ECNTBFn clock show the timing that time counter ECNTAFn and event counter ECNTBFn execute the count operation or clearing, respectively.

The operations of each register in up/down count operation mode are as follows:

- ECNTAFn: Measures time using one of the clock buses 0 to 5. When a compare match is detected, the count value is cleared synchronized with the next Pϕ clock.
- ECNTBFn: Upcount/downcount operation is performed at both rising and falling edges of TIFnA. Count direction is detedmined by the TIFnB input level. (See table 13.19.) Because of synchronization processing, a delay of two cycles occurs in TIFnA and TIFnB.
- GRAFn: Functions as the compare match register for ECNTAFn. A compare match is detected when the count number in ECNTAFn and GRAFn agree.
- GRBFn: Functions as the capture register for ECNTBFn. When a compare match in ECNTAFn is detected, this register captures the ECNTBFn count number synchronizing with the next Pϕ clock.
- ICFFn flag: After detecting a compare match in ECNTAFn, sets the ICFFn flag synchronized with the next Pϕ clock.
- ECNTCFn, GRCFn, GRDFn: Do not function

Table 13.19 Count Direction in Up/Down Event Count Mode

	Count Direction			
Input	**Upcount**		**Downcount**	
TIFnA	↑	↓	↑	↓
TIFnB	Low		High	

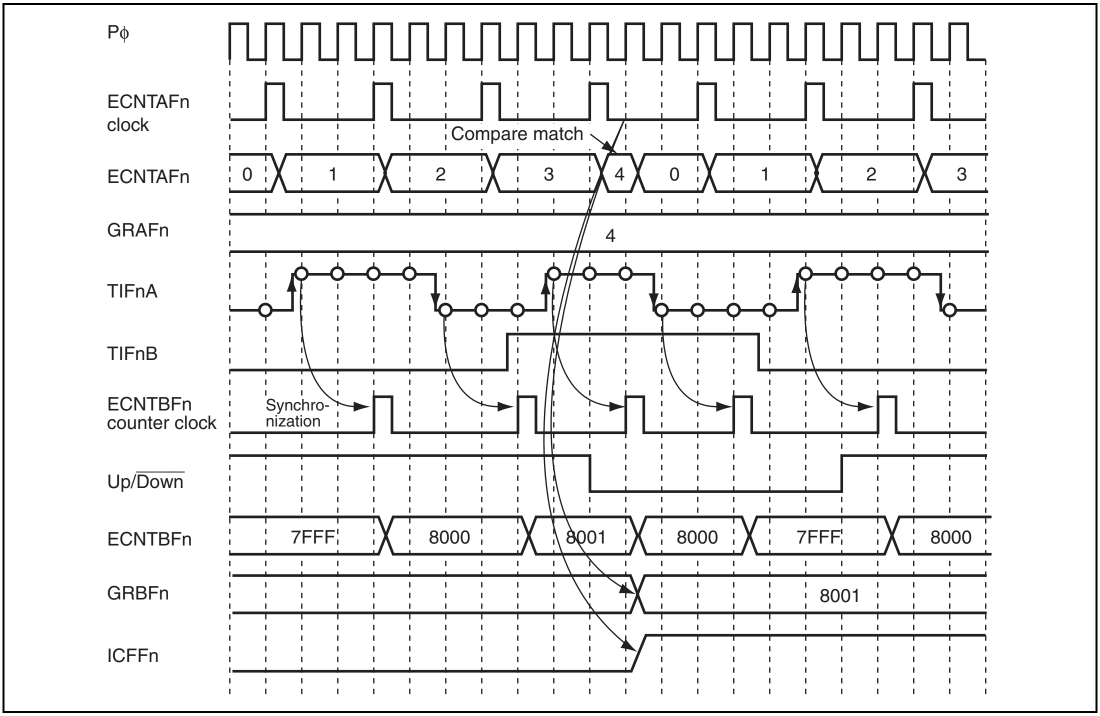


Figure 13.43 Operation Example of Up/Down Event Count

13.26.7 Four-time Multiplication Event Count

The count operation is executed using the external two input pins (TIFnA, TIFnB) as the count sources. Upcount or downcount is switched according to their input states. If a count period is designated to GRAFn, the count number after designation can be obtained in GRBFn. The counting period is the period of ECNTAFn cont source clock (GRAFn value).

At this moment, the timer F operates as shown below. Figure 13.44 shows an example of operation. The ECNTAFn clock and ECNTBFn clock show the timing that time counter ECNTAFn and event counter ECNTBFn execute the count operation or clearing, respectively.

The operations of each register in 'four-time event count operation' mode are as follows:

- ECNTAFn: Measures time using one of the clock buses 0 to 5. When a compare match is detected, the count value is cleared synchronized with the next Pϕ clock.
- ECNTBFn: Upcount/downcount operation is performed at both rising and falling edges of TIFnA and TIFnB respectively. Count direction is detedmined by the other signal input level. (See table 13.20). Because of synchronization processing, a delay of two cycles occurs in TIFnA and TIFnB.
- GRAFn: Functions as the compare match register for ECNTAFn. A compare match is detected when the count number in ECNTAFn and GRAFn agree.
- GRBFn: Functions as the capture register for ECNTBFn. When a compare match in ECNTAFn is detected, this register captures the ECNTBFn count number synchronizing with the next Pϕ clock.
- ICFFn flag: After detecting a compare match in ECNTAFn, sets the ICFFn flag synchronized with the next Pϕ clock.
- ECNTCFn, GRCFn, GRDFn: Do not function

Table 13.20 Count Direction in Four-time Multiplication Event Count Mode

	Count Direction							
Input	Upcount				Downcount			
TIFnA	High	↓	Low	↑	High	↓	Low	↑
TIFnB	↑	High	↓	Low	↓	Low	↑	High

Note: Operation when edge inputs in TIFnA and TIFnB are detected simultaneously is not guaranteed. The interval between edge inputs in TIFnA and TIFnB must be at least 1.5 cycles (Pϕ clock).

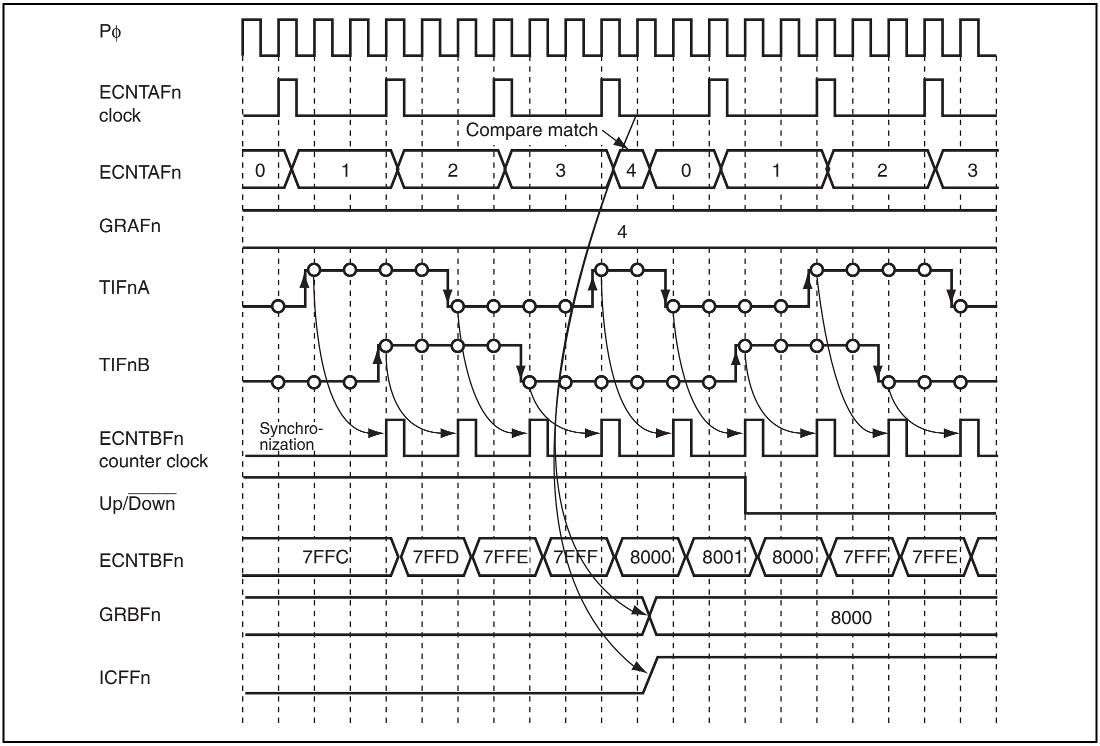


Figure 13.44 Operation Example of Four-time Multiplication Event Count Overflow/Underflow

When the count value H'FFFFFF00 (ECNTAFn, ECNTCFn) changes to H'00000000 (ECNTAFn, ECNTCFn), or H'FFFF (ECNTBFn) to H'0000 (ECNTBFn) in other than counter clear processing, an overflow will be detected. In this case, a flag is immediately set with the value reaching H'00000000 (or H'0000). (No figure is given here.) OVFAF, ECNTBFn, or OVFCn are set when an overflow is detected in ECNTAFn, ECNTBF, or ECNTCFn, respectively.

An underflow will be detected when the value H'0000 (ECNTBFn) changes to H'FFFF (ECNTBFn) and a flag is immediately set with the value reaching H'FFFF. (No figure is given here.) An underflow occurs only in ECNTBFn and OVFBn is set.

13.26.8 Overflow and Underflow

When a counter value changes H'FFFFFF00 (ECNTAFn, ECNTCFn) to H'00000000 (ECNTAFn, ECNTCFn) and H'FFFF (ECNTBFn) to H'0000 (ECNTBFn) except counter clear operation, overflow is detected. Overflow flags are set at the same time when the counter value changes H'FFFFFF00 (or H'0000). OVFAFn, OVFBFn, and OVFCFn are set when overflow is detected in ECNTAFn, ECNTBFn, and ECNTCFn, respectively.

When a counter value changes H'0000 (ECNTBFn) to H'FFFF (ECNTBFn), underflow is detected. Underflow flag is set at the same time when the counter value changes from H'0000 to H'FFFF. Underflow occurs only in ECNTBFn, and OVFBFn is set upon its detection.

13.27 Overview of Timer G

Timer G consists of six subblocks which are identical with each other.

An active-low pulse is output for one cycle of the Pϕ clock after a given time has elapsed. The time is counted. The generated pulse is used to activate the A/D converter or interrupt trigger.

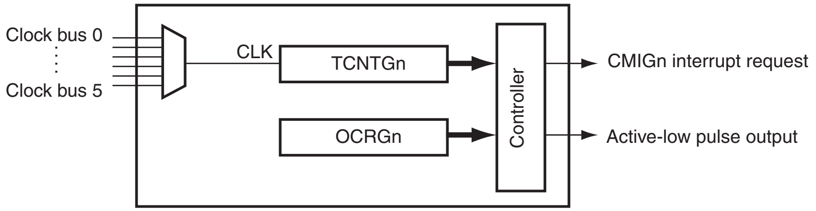
Interrupt requests can be issued other than the pulse signal and can request DMA transfer to the DMAC.

The counter clock is selected from six lines of the clock bus.

13.27.1 Block Diagram of Robots

Timer G subblocks consist of one 16-bit timer counter G (TCNTG) and one compare match register (OCRG), and controller.

Figure 13.45 is a block diagram of timer G.



[Legend]
TCNTG: Timer counter G
OCRG: Compare match register G
n = 0 to 5

Figure 13.45 Block Diagram of Timer G

13.27.2 Interrupt Requests

Six timer G interrupts, CMIG0 to CMIG5, are available in timer G. When a compare match is detected in the subblocks, an interrupt request is output. The interrupt request is received in the direct memory access controller (DMAC) and interrupt controller (INTC).

13.28 Description of Timer G Registers

13.28.1 Timer Start Register G (TSTRG)

TSTRG is an 8-bit readable/writable register that enables and disables the subblocks of timer G. Timer G counters run when the STRG bit and the TGE bit in the ATU-III master enable register (ATUENR) are both set to 1.

TSTRG can be read from and written to in byte or word units.

TSTRG is initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	-	-	STRG5	STRG4	STRG3	STRG2	STRG1	STRG0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	STRG5	0	R/W	Counter G Start These bits enable and disable timer counter Gn (TCNTGn) in the subblock. When these bits are cleared to 0, TCNTGn is stopped. While TCNTGn is stopped, it retains the previous value. When these bits are set to 1, TCNTGn is resumed from the previous value. TCNTGn runs when these bits and the TGE bit in ATUENR are both set to 1. 0: TCNTGn is disabled 1: TCNTGn is enabled The prescalers run regardless of the counter G start bit and are not synchronized with the timing at which TCNTG is started. Therefore, the time from when the counter G start bit is set to when TCNTG is incremented for the first time is less than the cycle of the clock of TCNTG.
4	STRG4	0	R/W	
3	STRG3	0	R/W	
2	STRG2	0	R/W	
1	STRG1	0	R/W	
0	STRG0	0	R/W	

Note: n = 0 to 5 (n denotes subblock number)

13.28.2 Timer Control Register G0 to G5 (TCRG0 to TCRG5)

TCRG0 to TCRG5 are 8-bit readable/writable registers that set the operating mode of each subblock of timer G.

TCRG0 to TCRG5 can be read from and written to in byte or word units.

TCRG0 to TCRG5 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	-	CKSELGn[2:0]			-	-	CMP OEGn	CM EGn
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R	R/W	R/W

Note: n = 0 to 5 (correspond to subblocks G0 to G5)

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6 to 4	CKSELGn [2:0]	000	R/W	Clock Select Gn These bits select the clock source of timer counter Gn (TCNTGn) of subblock. However, do not set to B'110 or B'111. If set, operation cannot guaranteed. 000: Clock-bus line 0 001: Clock-bus line 1 010: Clock-bus line 2 011: Clock-bus line 3 100: Clock-bus line 4 101: Clock-bus line 5 110: Setting prohibited 111: Setting prohibited
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
1	CMPOEGn	0	R/W	Pulse Output Enable Gn Selects whether or not a compare match pulse is externally output on compare match between timer counter Gn (TCNTGn) and compare match register (OCRGn). 0: Pulse is not output on compare match between TCNTGn and OCRGn 1: Pulse is output on compare match between TCNTGn and OCRGn
0	CMEGn	0	R/W	Compare Match Interrupt Enable Gn Enables and disables output of interrupt requests on compare match between timer status register Gn (TSRGn) and compare match flag Gn (CMFGn). 0: Interrupt request is not issued on compare match of CMFGn 1: Interrupt request is issued on compare match of CMFGn

13.28.3 Timer Status Registers G0 to G5 (TSRG0 to TSRG5)

TSRG0 to TSRG5 are 8-bit readable/writable registers that measure the time and indicate occurrence of event counter overflow and compare match.

These flags are interrupt request sources and interrupt requests for the CPU or DMA transfer requests for the DMAC can be issued when bits in timer control registers G0 to G5 (TSRG0 to TSRG5) are set to 1.

TSRG0 to TSRG5 can be read from and written to in byte or word units.

TSRG0 to TSRG5 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	OVFGn	CMFGn
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/(W)*	R/(W)*

Note: n = 0 to 5 (correspond to subblocks G0 to G5)
* Only 0 can be written to this bit after it is read as 1 to clear it. Writing 1 is ignored.

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	OVFGn	0	R/(W)*	Overflow Flag Gn Indicates whether or not timer counter Gn (TCNTGn) has overflowed. This bit cannot be set to 1 by software. No interrupt corresponds to this bit. 0: TCNTGn has not overflowed [Clearing condition] • When writing 0 to this bit after reading it as 1 1: TCNTGn has overflowed [Setting condition] • When TCNTGn overflowed (from H'FFFF to H'0000)
0	CMFGn	0	R/(W)*	Compare Match Flag Gn Indicates whether or not a compare match has occurred in subblocks G0 to G5. This flag cannot be set to 1 by software. When the CMEGn bit in the timer control register is set to 1 and this flag is set to 1, a compare match interrupt is issued. 0: Compare match has not occurred in subblocks Gn [Clearing conditions] • When writing 0 to this bit after reading it as 1 • When DMAC received transfer request by compare match interrupt 1: Compare match has occurred in subblocks Gn [Setting condition] • When compare match occurred in subblocks Gn

13.28.4 Timer Counters G0 to G5 (TCNTG0 to TCNTG5)

TCNTG0 to TCNTG5 are 16-bit readable/writable registers. These registers are provided one for each subblock and are incremented by the clock selected in the corresponding control register. Lines 0 to 5 of the clock bus can be selected as the input clock.

These counter values are constantly compared with the value in compare match register G (OCRG). When they match, Compare match flag G (CMFG) is set and the counter value is cleared to H'0000 in the next Pϕ clock cycle. If counter clearing by compare match and incrementation occur simultaneously, TCNTG is initialized to H'0001. This occurs when TCNTG is driven by the clock whose frequency is equal to the Pϕ clock.

TCNTG0 to TCNTG5 can be read from and written to in word units.

TCNTG0 to TCNTG5 are initialized to H'0000 by a power-on reset or a transition to the hardware standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TCNTGn[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: n = 0 to 5 (correspond to subblocks G0 to G5)

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	TCNTGn [15:0]	All 0	R/W	Timer Counter Gn These bits store the up-counter value.

13.28.5 Compare Match Registers G0 to G5 (OCRG0 to OCRG5)

OCRG0 to OCRG5 are 16-bit readable/writable registers. These registers are provided one for each subblock and function as the output compare register for timer counter G (TCNTG).

Do not set OCRG to H'0000. If H'0000 is set, compare matches occur at unwanted cycles.

OCRG0 to OCRG5 can be read from and written to in word units.

OCRG0 to OCRG5 are initialized to H'FFFF by a power-on reset or a transition to the hardware standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OCRGn[15:0]															
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: n = 0 to 5 (correspond to subblocks G0 to G5)

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	OCRGn [15:0]	All 1	R/W	Compare Match Gn These bits set the compare match value.

13.29 Operations of Timer G

An active-low pulse is output for one cycle of the Pϕ clock after a time set in OCRG has elapsed. The initial level on the output pin is a level of 1. Set the number of cycles of the TCNTG clock in OCRG.

The generated pulse can be used to activate the A/D converter by setting the compare match pulse output enable bit (CMPOEG) in timer control register G (TCRG).

When compare match occurs, the compare match flag (CMFG) in timer status register G (TSRG) is set. DMA transfer or interrupt requests can be issued for the DMAC or CPU by setting the compare match interrupt enable bit (CMEG) in TCRG.

Figure 13.46 shows operation example for counters and compare match. In this example, the TCNTG clock is an ideal signal to show count timing or clearing timing.

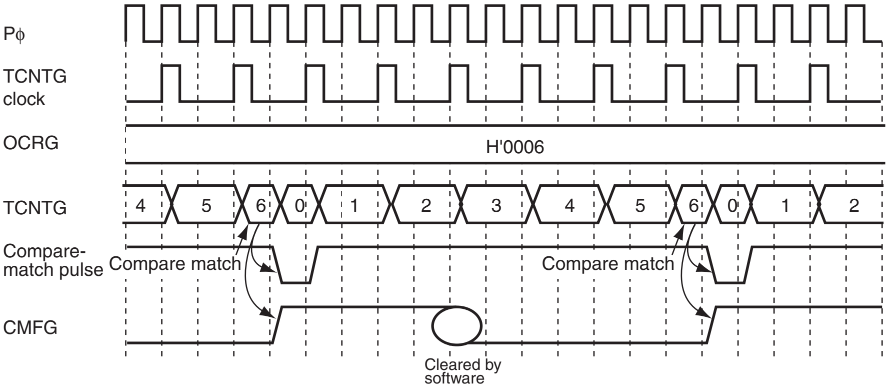


Figure 13.46 Operation Example of Counter and Compare Match

13.30 Overview of Timer H

Timer H is a counter that measures a given time repeatedly.

Timer counter 1H (TCNT1H) is a 16-bit up-counter driven by the clock selected from six lines of the clock bus. When the value in TCNT1H reaches the value in compare register 1H (OCR1H), it is cleared to H'0000 and is incremented again.

Timer counter 2H (TCNT2H) is a 32-bit counter that is incremented on compare match between TCNT1H and OCR1H, meaning that the counter counts the compare match.

Interrupt requests can be issued when TCNT2H is incremented.

13.30.1 Block Diagram of Timer H

Timer H consists of one 16-bit timer counter 1H (TCNT1H), one compare match register 1H (OCR1H), one 32-bit timer counter 2H (TCNT2H), and controller.

Figure 13.47 is a block diagram of timer H.

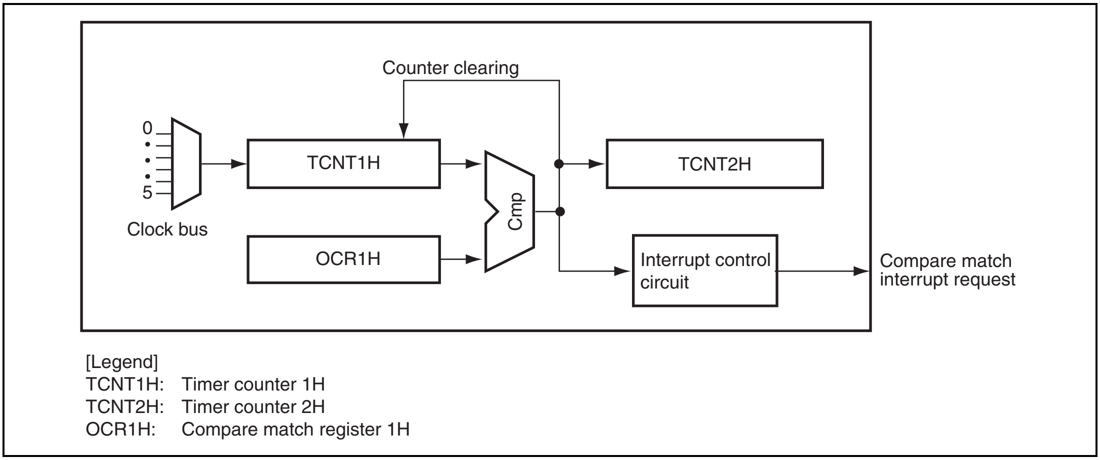


Figure 13.47 Block Diagram of Timer H

13.30.2 Interrupts

Timer H has one interrupt request signal (CMIH). Interrupt requests are issued on compare match between TCNT1H and OCR1H.

13.31 Description of Timer H Registers

13.31.1 Timer Control Register H (TCRH)

TCRH is an 8-bit readable/writable register that selects the counter clock and controls output of compare match interrupt.

TCRH can be read from and written to in byte or word units.

TCRH is initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	-	CKSELH[2:0]			-	-	-	CMEH
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6 to 4	CKSELH [2:0]	000	R/W	Clock Select H These bits select the source of the clock of timer counter 1H (TCNT1H). 000: Clock-bus line 0 001: Clock-bus line 1 010: Clock-bus line 2 011: Clock-bus line 3 100: Clock-bus line 4 101: Clock-bus line 5 110: Setting prohibited 111: Setting prohibited
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
0	CMEH	0	R/W	Compare Match Interrupt Enable H Enables and disables the interrupt request for compare match flag H (CMFH) in timer status register H (TSRH). 0: CMFH interrupt is disabled 1: CMFH interrupt is enabled

13.31.2 Timer Status Register H (TSRH)

TSRH is an 8-bit readable/writable register that indicates occurrence of compare match between timer counter 1H (TCNT1H) and the compare match register (OCR1H), TCNT1H overflow, and overflow on timer counter 2H (TCNT2H).

The compare match flag is an interrupt source and can be used for output of the interrupt request when the bit in timer control register F (TCRH) is set to 1.

TSRH can be read from and written to in byte or word units.

TSRH is initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	OVF2H	OVF1H	CMFH
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/(W)*	R/(W)*	R/(W)*

Note: * Only 0 can be written to this bit after it is read as 1 to clear it. Writing 1 is ignored.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
2	OVF2H	0	R/(W)*	Overflow Flag 2H Indicates whether or not timer counter 2H (TCNT2H) has overflowed. This flag cannot be set to 1 by software. No interrupt corresponds to this flag. 0: TCNT2H has not overflowed [Clearing condition] • When writing 0 to this bit after reading it as 1 1: TCNT2H has overflowed [Setting condition] • TCNT2H has overflowed (from H'FFFFFFFF to H'00000000) Writing H'00000000 to TCNT2H or starting TCNT2H up from the initial value (H'00000000) has no effect on this bit When writing a value to the counter and incrementation occur simultaneously while the counter is H'FFFFFFFF, the overflow flag is set to 1 but the counter value is changed to the written value instead of H'0000.
1	OVF1H	0	R/(W)*	Overflow Flag 1H Indicates whether or not timer counter 1H (TCNT1H) has overflowed. This flag cannot be set to 1 by software. No interrupt corresponds to this flag. 0: TCNT1H has not overflowed [Clearing condition] • When writing 0 to this bit after reading it as 1 1: TCNT1H has overflowed [Setting condition] • TCNT1H has overflowed (from H'FFFF to H'0000) Writing H'0000 to TCNT1H or starting TCNT1H up from the initial value (H'0000) has no effect on this bit When writing a value to the counter and incrementation occur simultaneously while the counter is H'FFFF, the overflow flag is set to 1 but the counter value is changed to the written value instead of H'0000.

Bit	Bit Name	Initial Value	R/W	Description
0	CMFH	0	R/(W)*	Compare Match Flag H Indicates whether or not compare match between TCNT1H and OCR1H has occurred. This flag cannot be set to 1 by software. When the CMEH bit in TCRH is set to 1, a compare match interrupt is issued by setting this flag. 0: Compare match between TCNT1H and OCR1H has not occurred [Clearing condition] • When writing 0 to this bit after reading it as 1 1: Compare match between TCNT1H and OCR1H has occurred [Setting condition] • When the values in TCNT1H and OCR1H match While CMFH is set to 1, meaning that the flag has not been cleared, the next compare match can be detected. In this case, 1 is rewritten to this bit.

Note: * Only 0 can be written to this bit after it is read as 1 to clear the flag. Writing 1 to this bit is ignored.

13.31.3 Timer Counter 1H (TCNT1H)

TCNT1H is a 16-bit readable/writable register. TCNT1H run when the THE bit in the ATU-III master enable register (ATUENR) and is incremented by the clock selected in the CKSEL bit in timer control register H (TCRH).

Overflow flag 1H (OVF1H) in timer status register H (TSRH) is set to 1 when TCNT1H overflows (from H'FFFF to H'0000)

The value in TCNT1H is constantly compared with the value in compare match register 1H (OCR1H). When they match, the CMFH bit in TSRH is set to 1 and TCNT1H is cleared to H'0000 in the next Pϕ clock cycle.

However, if counter clearing by compare match and incrementation occur simultaneously, TCNT1H is cleared to H'0001. This occurs when TCNT1H is driven by the clock whose frequency is equal to the Pϕ clock.

At the same time as compare match, timer counter 2H (TCNT2H) is incremented.

TCNT1H can be read from and written to in word units.

TCNT1H is initialized to H'0000 by a power-on reset or a transition to the hardware standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TCNT1H[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	TCNT1H [15:0]	All 0	R/W	Timer Counter 1H These bits store the 16-bit up-counter value.

13.31.4 Compare Match Register 1H (OCR1H)

OCR1H is a 16-bit readable/writable register that function as the output compare register for timer counter 1H (TCNT1H). Compare match occurs at the following cycles.

Cycle of compare match =

cycle of the TCNT1H counter clock (selected in the CKSELH bit in TCRH) × value in OCR1H

TCNT2H is incremented at this cycle. Interrupt requests are output when they are enabled,

Do not set OCR1H to H'0000. If set, compare match occurs at unwanted cycles.

OCR1H can be read from and written to in word units.

OCR1H is initialized to H'FFFF by a power-on reset or a transition to the hardware standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OCR1H[15:0]															
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	OCR1H [15:0]	All 1	R/W	Compare Match 1H These bits store the compare match value.

13.31.5 Timer Counter 2H (TCNT2H)

TCNT2H is a 32-bit readable/writable register. TCNT2H is incremented every compare match between timer counter 1H (TCNT1H) and compare match register 1H (OCR1H).

TCNT2H runs when the THE bit in the ATU-III master enable register (ATUENR).

When TCNT2H overflows (from H'FFFF FFFF to H'0000 0000), overflow flag 2H (OVF2H) in timer status register H (TSRH) is set to 1.

TCNT2H can be read from and written to in longword units.

TCNT2H is initialized to H'0000 0000 by a power-on reset or a transition to the hardware standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TCNT2H[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TCNT2H[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TCNT2H [31:0]	All 0	R/W	Timer Counter 2H These bits store the 32-bit counter value.

13.32 Operations of Timer H

Compare match occurs and the compare match flag (CMFH) is set to 1 when a time set in the compare match register (OCR1H) has elapsed. At this time, TCNT2H is incremented and TCNT1H is cleared to H'0000.

The TCNT1H counter clock is selected by the CKSELH bit in TCRH. TCNT1H and TCNT2H run when the THE bit in ATUENR is set to 1. If the THE bit is cleared to 0 while the counters are in operation, TCNT1H and TCNT2H are stopped and retain the counter value unchanged. When the THE bit is set to 1, the counters are resumed from the retained value.

Interrupt requests can be issued depending on the compare match interrupt enable H bit (CMEH).

Figure 13.48 shows an operation example of timer H. In this example, the TCNT1H counter clock is an ideal signal to show counter operation or clearing operation timing.

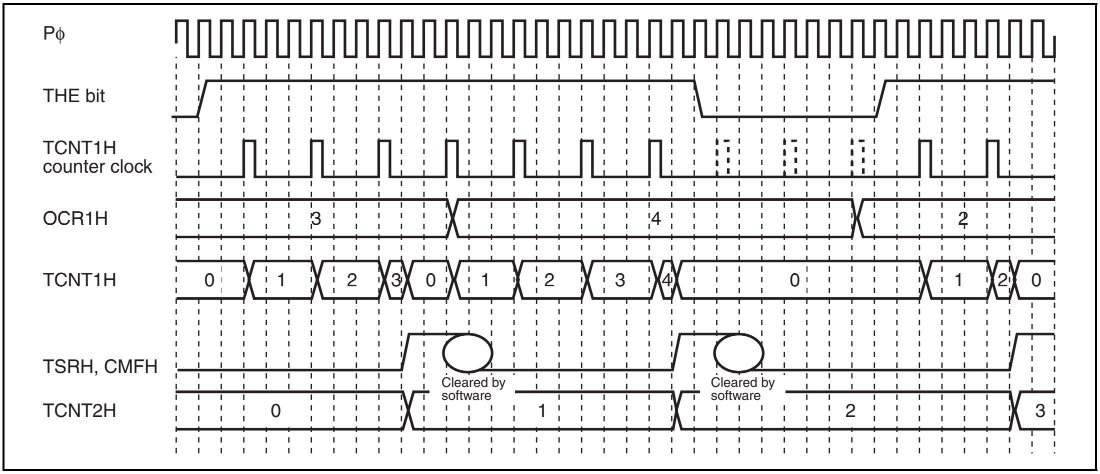


Figure 13.48 Operations of Timer H

13.33 Overview of Timer J

Timer J consists of two identical subblocks that measure a given time repeatedly.

Timer counter J (TCNTJ) is a 16-bit counter and is incremented by the clock selected from six lines of the clock bus. TCNTJ is cleared by the input edge of the TIJ pin. Timer J also has nine 16-bit FIFO registers which latch the value in TCNTJ every edge input. When the FIFO becomes full, DMAC activation or an interrupt request can be issued.

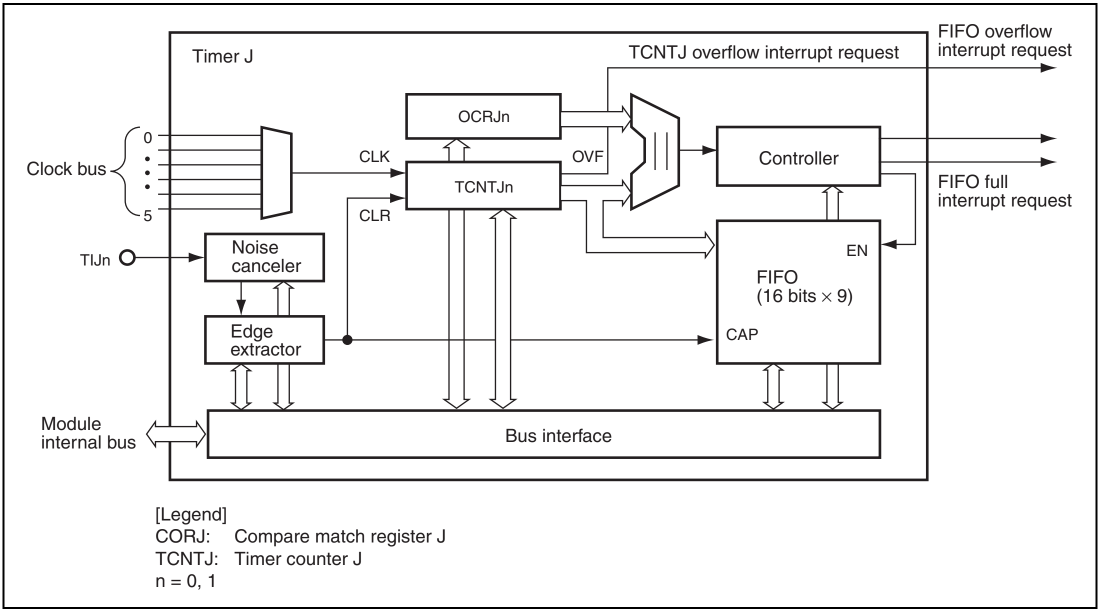
Timer J controls the valid period of the FIFO by controlling the time from compare match of compare match register J (OCRJ) to FIFO full.

Noise of the externlly input signals (TIJ) can be removed by the input noise cancelltion function.

13.33.1 Block Diagram of Timer J

Timer J consists of two identical subblocks. Each subblock includes one 16-bit timer counter J (TCNTJ), one compare match register J (OCRJ), nine 16-bit FIFO registers, input signal controller (edge extractors and noise cancelers), and controller.

Figure 13.49 is a block diagram of timer J.



[Legend]
CORJ: Compare match register J
TCNTJ: Timer counter J
n = 0, 1

Figure 13.49 Block Diagram of Timer J

13.34 Description of Timer J Registers

13.34.1 Timer Start Register J (TSTRJ)

TSTRJ is an 8-bit readable/writable register that enables and disables the subblocks (timer J0 and timer J1) of timer J. Timer J counters run when the counter J start bit (STRJ) and timer J enable bit (TJE) in the ATU-III master enable register (ATUENR) are both set to 1.

The prescalers run regardless of the counter J start bit and are not sychronized with the timing at which the TCNTJn is started. Therefore, the time from when the THE bit is set to when TCNTJn is incremented for the first time is less than the cycle of the clock of TCNTJn.

TSTRJ can be read from and written to in byte or word units.

TSTRJ is initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	STRJ1	STRJ0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	STRJ1	0	R/W	Counter Jn Start These bits enable and disable timer counter Jn (TCNTJn). When these bits are cleared to 0, TCNTJn is stopped. While TCNTJn is stopped, it retains the previous value. When these bits are set again, the counter is resumed from the value. 0: Counting of TCNTJn is stopped 1: Counting of TCNTJn is enabled
0	STRJ0	0	R/W	

Note: n = 0, 1

13.34.2 Timer Control Registers J0 and J1 (TCRJ0 and TCRJ1)

TCRJ0 and TCRJ1 are 8-bit readable/writable registers that select operation modes of each subblock (timer J0 and timer J1)

TCRJ0 and TCRJ1 can be read from and written to in byte or word units.

TCRJ0 and TCRJ1 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	-	CKSELJn[2:0]			-	NCEJn	IOJn[1:0]	
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Note: n = 0, 1

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6 to 4	CKSELJn [2:0]	000	R/W	Clock Select Jn These bits select the clock source of timer counter Jn (TCNTJn) of subblock. However, do not set to B'110 or B'111. If set, operation cannot guaranteed. 000: Clock-bus line 0 001: Clock-bus line 1 010: Clock-bus line 2 011: Clock-bus line 3 100: Clock-bus line 4 101: Clock-bus line 5 110: Setting prohibited 111: Setting prohibited
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
2	NCEJn	0	R/W	Noise Canceler Enable Jn These bits enable and disable the noise cancelers for externally input signals (TIJn) When a level change on externally input signal TIJn is detected while this bit is set to 1, it is processed in premature-transition cancellation or minimum time-at-level cancellation mode depending on the setting in the noise cancelltion mode register (NCMR) of the common controller. • In premature-transition cancellation mode When a level change of the externally input signal is detected, the change is output as the signal whose noise is removed and the corresponding noise canceler counter (NCNTJn) is started for counting up. Subsequent level changes are masked until the value in the counter reaches the value in the noise canceler register (NCRJn). The level of the externally input signal is output on this compare match. When these bits are cleared to 0 while the counter (NCNTJn) is being incremented, counting continues until the values in the counter and the noise canceler register match. The subsequent level changes are also masked over this period. • In minimum time-at-level cancellation mode When a level change of the externally input signal is detected, the corresponding noise canceler counter (NCNTJn) is started for counting up. If subsequent level changes are not detected until the value in the counter reaches the value in the noise canceler register (NCRJn), the previously accepted level change is output as the signal whose noise is removed on compare match of the counter and noise canceler register. When the subsequent level change is detected before the values in the counter and noise canceler register match, all the changes are treated as noise. Therefore the signal whose noise is removed is not changed. When these bits are cleared to 0 while the counter (NCNTJn) is being incremented, counting continues until the values in the counter and the noise canceler register match or a level change on the externally input signal is detected. For details on operations in each mode, see figures 13.1 and 13.2. 0: Noise canceler for TIJn is disabled 1: Noise canceler for TIJn is enabled

Bit	Bit Name	Initial Value	R/W	Description
1, 0	IOJn[1:0]	00	R/W	I/O Control Jn These bits select the edge of external inputs (TIJn) that is to be extracted for use in input-capture triggering. When an edge selected in these bits is detected, the contents of timer counter Jn (TCNTNJn) are transferred to FIFO J (FIFOJn). Edges are extracted from the signal affter noise removal. When the noise canceler is disabled, the selected edge is simply extracted from the external inputs (TIJn). When the noise canceler is enabled, the selected edge is extracted from the signals after noise removal. Edge extraction is synchronized with the Pϕ clock. Make sure that the frequency of the Pϕ clock is at least twice the frequency of the external input signal. Otherwise, edge extraction will not performed correctly. 00: Input capturing is not performed 01: TCNTJ is captured in ICRJ on the rising edge of TIJ 10: TCNTJ is captured in ICRJ on the falling edge of TIJ 11: TCNTJ is captured in ICRJ on both edges of TIJ

Note: n = 0, 1

13.34.3 FIFO Control Registers J0 and J1 (FCRJ0 and FCRJ1)

FCRJ0 and FCRJ1 are 8-bit readable/writable registers that control FIFO of each subblock (timer J0 and timer J1).

FCRJ0 and FCRJ1 can be read from and written to in byte or word units.

FCRJ0 and FCRJ1 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	FIFO ENJn	-	FVCR ENJn	FRS TJn	-	-	FDFTR GJn[1:0]	
Initial value:	0	0	0	0*	0	0	0	0
R/W:	R/W	R	R/W	W	R	R	R/W	R/W

Note: n = 0, 1

* Writing 0 is ignored. This bit is always read as 0 even if 1 is written.

Bit	Bit Name	Initial Value	R/W	Description
7	FIFOENJn	0	R/W	FIFO Register Enable Jn Enables and disables FIFO register Jn (FIFOJn) in subblocks J0 and J1. When this bit is set to 1 and the FIFO is disabled, even if an edge is input on pin TIJn, the value in TCNTJn is not latched in FIFO. However when FVCRENJn is set to 1, even if this bit is 0, the value in TCNTJn is latched in the FIFO every input edge until FIFO becomes full after compare match on OCRJn. 0: FIFO Jn is disabled 1: FIFO Jn is enabled
6	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
5	FVCRENJn	0	R/W	FIFO Enable Control Enable Jn When FIFOENJn = 0 and the FIFO is disabled, the FIFO is temporarily enabled until it becomes full (the threshold level can be set by FDFTRGJ) after compare match of compare match register Jn (OCRJn). Note that the FIFO is always enabled regardless of this bit when bit FIFOENJn is set to 1. 0: FIFO is not enabled on OCRJn compare match 1: FIFO is enabled until it becomes full after OCRJn compare match
4	FRSTJn	0*[1]	W	FIFO Data Register Reset Jn Data captured in the FIFO is discarded and then the FIFO becomes empty (FIFO reset). However, the FDFFJn and FDOVFJn flags which have been set to 1 are not cleared. 0: No operation 1: FIFO is reset
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	FDFTRGJn [1:0]	00	R/W	FIFO Data Full Trigger Jn These bits set the number of data words which is the threshold for setting the FDFFJn flag in the timer J status register (TSRJn). When the FIFO is enabled and the number of data words in it has reached the trigger count, the FDFFJn bit is set to 1. When the FIFO is enabled on compare match while FIFOENJn = 0 and FVCRENJn = 1, and then the number of data words has reached the trigger count, the FIFO is disabled. 00: 9 01: 6 10: 4 11: 2

Notes: n = 0, 1

1. Writing 0 to this bit is ignored. The write data is not held. This bit is always read as 0.
2. When the setting of FIFOENJn is 0 and that of FVCRENJn is 1, FDFTRGJn[1:0] can only be set to 00 when a FIFO data full interrupt is generated for the CPU. After generation of the FIFO data full interrupt, the service routine should read all data from the FIFO, use the FRSTJn bit to reset the FIFO registers, and then clear FDFFJn to 0.

13.34.4 Timer Status Register J0 and J1 (TSRJ0 and TSRJ1)

TSRJ0 and TSRJ1 are 8-bit readable/writable registers that indicate occurrence of overflow on timer counter Jn (TCNTJn), and compare match on compare register Jn (OCRJn). In addition, there are flags that indicate that the FIFO has overflowed and the number of data in the FIFO exceeds the trigger count.

The flags other that FIFO status flag Jn (FVLDFJn) and compare match flag Jn (CMFJn) are used for generating interrupt requests. When the corresponding bit in timer interrupt enable register Jn (TIERJn) is set to 1, DMA transfer requests for the DMAC or interrupt request for the CPU can be issued.

TSRJ0 and TSRJ1 can be read from and written to in byte or word units.

TSRJ0 and TSRJ1 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	FVLD FJn	CMF Jn	OVFJn	FDOV FJn	FDF FJn
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: n = 0, 1

* Only 0 can be written to this bit after it is read as 1 to clear it. Writing 1 is ignored.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
4	FVLDFJn	0	R	FIFO Status Flag Jn Indicates whether the FIFO is enabled or disabled. This bit cannot be set or cleared by software. When this bit is 1, the corresponding FIFO Jn (FIFOJn) is enabled. The value in TCNTJn is latched in the FIFO on the edge of the TIJn pin. When FIFOENJn = 1, this bit is always set to 1. When FIFOENJn = 0 and FVCRENJn = 1, FVLDFJn becomes 1 on OCRJn compare match. When the FIFO becomes full, FVLDFJn is cleared to 0. When FIFOENJn = 0 and FVCRENJn = 0, this bit is always 0. 0: FIFO is not enabled [Clearing conditions] • When FIFOENJn is cleared to 0 • When FIFOENJn is 0, FVCRENJn is 1, and the number of data words in FIFO exceeds the value set in FDFTRGJn 1: FIFO is enabled [Setting conditions] • When FIFOENJn is set to 1 • When FIFOENJn is 0, FVCRENJn is 1, and OCRJn compare match occurred
3	CMFJn	0	R/(W)*	Compare Match Flag Jn Indicates that compare match between TCNTJn and OCRJn has occurred. When this bit is 1, compare match on compare match register Jn (OCRJn) has occured. This bit cannot be set to 1 by software. 0: Compare match between OCRJn and TCNTJn has not occurred [Clearing condition] • When writing 0 to this bit after reading it as 1 1: Compare match between OCRJn and TCNTJn has occurred [Setting condition] • When the values in TCNTJn and OCRJn match

Bit	Bit Name	Initial Value	R/W	Description
2	OVFJn	0	R/(W)*	Overflow Flag Jn Indicates that timer counter Jn (TCNTJn) has overflowed. When this bit is 1, timer counter Jn (TCNTJn) has overflowed. This bit cannot be set to 1 by software. 0: TCNTJn has not overflowed [Clearing condition] • When writing 0 to this bit after reading it as 1 1: TCNTJn has overflowed [Setting condition] • When TCNTJn has overflowed (from H'FFFF to H'0000)
1	FDOVFJn	0	R/(W)*	FIFO Data OVerflow Flag Jn Indicates that another capture has occurred while nine words of data was in the FIFO. When this bit is 1, FIFOJn has overflowed and captured data is lost. This bit cannot be set to 1 by software. 0: Data in the FIFO is normal [Clearing condition] • When writing 0 to this bit after reading it as 1 1: Another capture has occurred while nine words of data was in the FIFO [Setting condition] • When another capture has occurred while nine words of data was in the FIFO

Bit	Bit Name	Initial Value	R/W	Description
0	FDFFJn	0	R/(W)*	FIFO Data Full Flag Jn Indicates that the number of data words (TCNTJn counter values) captured in the FIFO is equal to or exceeds the value set in the FDFTRGJn bits in the FIFO control register Jn (FCRJn). When this bit is 1, the number of data words captured in the FIFO is equal to or exceeds the set value and the captured data is ready to be read. This bit cannot be set to 1 by software. To clear the flag, write 0 to this bit after reading it as 1. However, the data in FIFOJn must be read to make the number of data words in the FIFO less than the value set in the FDFTRGJn bits. Writing 0 before reading it as 1 is ignored. When data is read by DMA transfer, this bit is cleared to 0. The transfer request is issued by the FIFO data full interrupt. However, if the number of data words in the FIFO is still equal to or exceeds the value set in the FDFTRGJn bits after the reading by DMA transfer, this bit is set to 1 again. 0: Number of data words in FIFO is less than the value in FDFTRGJn [Clearing conditions] • When writing 0 to this bit after reading it as 1 • When DMAC receives DMA transfer request by FIFO data full interrupt 1: Number of data words in FIFO is equal to or exceeds the value in FDFTRGJn [Setting condition] • When the number of data words in FIFO has reached the value in FDFTRGJn

Notes: n = 0, 1

* Only 0 can be written to this bit after it is read as 1 to clear the flag. Writing 1 to this bit is ignored.

13.34.5 Timer Interrupt Enable Registers J0 and J1 (TIERJ0 and TIERJ1)

TIERJ0 and TIERJ1 are 8-bit readable/writable registers that enable and disable interrupt requests for the status flag in timer status register J (TSRJ). Data in the FIFO can be read with DMA transfer requested by the FIFO data full interrupt.

TIERJ0 and TIERJ1 can be read from and written to in byte or word units.

TIERJ0 and TIERJ1 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	OV EJn	FDOV EJn	FDF EJn
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

Note: n = 0, 1

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	OVEJn	0	R/W	Overflwo Interrupt Enable Jn Enables and disables interrupt requests for the status flag (OVEJn) of overflow on timer counter Jn (TCNTJn). 0: OVEJn interrupt request is disabled 1: OVEJn interrupt request is rnabled
1	FDOVEJn	0	R/W	FIFO Data Overflow Interrupt Enable Jn Enables and disables interrupt requests for the status flag (FDOVEJn) of overflow on the FIFO register Jn (FIFOJn) 0: FDOVEJn interrupt request is disabled 1: FDOVEJn interrupt request is enabled
0	FDFEJn	0	R/W	FIFO Data Full Interrupt Enable Jn Enables and disables interrupt requests for the status flag (FDFEJn) of data full of the FIFO register Jn (FIFOJn). While the interrupt request is enabled, set the DMAC. DMA transfer can be activated by the interrupt request. 0: FDFEJn interrupt request is disabled 1: FDFEJn interrupt request is enabled

13.34.6 Timer Counter J0 and J1 (TCNTJ0 and TCNTJ1)

TCNTJ0 and TCNTJ1 are 16-bit readable/writable reigsters. These registers are provided one for each subblock and are incremented by the clcok selected from lines 0 to 5 of the clock bus depending on the control register. The counter value is cleared to H'0000 by the edge of pin TIJn.

TCNTJ0 and TCNTJ1 can be read from and wirtten to in word units.

TCNTJ0 and TCNTJ1 are initialized to H'0000 by a power-on reset or a transition to the hardware standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TCNTJn[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: n = 0, 1

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	TCNTJn [15:0]	All 0	R/W	Timer Counter Jn These bits store the up-counter value.

13.34.7 Compare Match Registers J0 and J1 (OCRJ0 and OCRJ1)

OCRJ0 and OCRJ1 are 16-bit readable/writable reigsters. These registers are provided one for each subblock and function as output compare registers for timer counter Jn (TCNTJn).

When FIFOENJn and FVCRENJn in FIFO control register Jn (FCRJn) is 0 and 1, respectively. The FIFO is enabled (FVLDFJn is set to 1) on OCRJn compare match.

OCRJ0 and OCRJ1 can be read from and wirtten to in word units.

OCRJ0 and OCRJ11 are initialized to H'FFFF by a power-on reset or a transition to the hardware standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OCRJn[15:0]															
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: n = 0, 1

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	OCRJn [15:0]	All 1	R/W	Compare Match Jn These bits store the compare match value.

13.34.8 FIFO Registers J0 and J1 (FIFOJ0 and FIFOJ1)

FIFOJ0 and FIFOJ1 are 16-bit read-only registers. These register are provided one for each subblock and can store nime words of data of timer counter Jn (TCNTJn). FIFOJn can be read but cannot be written by the CPU. If FIFOJn in which no data is captured is read, the read value is undefined.

If another capture occurs while nine words of data is cpatured in the FIFO, the latest data is lost. Secure the free area by reading the FIFO before another capture has occured.

FIFOJ0 and FIFOJ1 can be read in word units.

FIFOJ0 and FIFOJ1 are initialized to H'0000 by a power-on reset or a transition to the hardware standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FIFODJn[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note: n = 0, 1

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	FIFODJn [15:0]	All 0	R	FIFO Data Jn These bits store the FIFO register data.

13.34.9 FIFO Data Count Registers J0 and J1 (FDNRJ0 and FDNRJ1)

FDNRJ0 and FDNRJ1 are 8-bit read-only registers. These registers are provided one for each subblock and indicate the number of data words in FIFOJ which stores captured data. This register cannot be written to.

This register is incremented by 1 every capture in FIFO and is decremented by 1 every read from FIFO (one-word read).

When the FIFO is reset by the FRSTJn bit in FIFO control register Jn (FCRJn), the value in this register is also cleared to H'0.

FDNRJ0 and FDNRJ1 can be read from in byte units.

FDNRJ0 and FDNRJ1 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	FDNJn[3:0]			
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Note: n = 0, 1

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3 to 0	FDNJn[3:0]	0000	R	FIFO Data Count Jn These bits indicate the number of data words captured in the FIFO. The value ranges from H'0 to H'9. H'0 indicates no data is in the FIFO and H'9 indicates that FIFOJn is full with the captured data.

13.34.10 Noise Canceler Counters J0 and J1 (NCNTJ0 and NCNTJ1)

NCNTJ0 and NCNTJ1 are 8-bit readable/writable registers.

These registers are incremented by an assertion of the input signals (TIJ0 and TIJ1) as triggers when the noise canceler function is enabled by the noise canceler enable bits (NCEJ1 and NCEJ0) in timer control register J (TCRJn). These registers are driven by the noise canceler counter clock, which is supplied by the prescaler.

Premature-transition cancellation or minimum time-at-level cancellation is performed depending on the setting in the noise cancelltion mode register (NCMR) of the common controller.

- In premature-transition cancellation mode

  When the NCEJn bit is 1, NCNTJn is stopped, and a level change on pin TIJn is detected, NCNTJn is started for counting up. The counter is cleared to H'00 and stopped on the first edge of the Pϕ clock after the counter value maches the value in noise canceler register Jn (NCRJn).

  NCNTJn is incremented regardless of the TJE bit in the ATU-III master enable register (ATUENR).

  The level change at the start of counting is output as the signal after noise removal and its edge is to be extracted. However, since subsequent level changes are masked, the noise removal signal is not changed. When the values in the counter and NCRJn match, the input signla level at this time is output as the noise removal signal.

  When NCEJn bits are cleared to 0 while the counter is being incremented, counting continues until the values in the counter and the noise canceler register match. The subsequent level changes are also masked over this period.

- In minimum time-at-level cancellation mode

  When the NCEJn bit is 1, NCNTJn is stopped, and a level change on pin TIJn is detected, NCNTJn is started for counting up. If subsequent level change is detected or the values in the counter and the noise canceler register Jn (NCRJn) match, the counter is cleared to H'00 and stopped on the next Pϕ clock cycle.

  NCNTJn is incremented regardless of the TJE bit in the ATU-III master enable register (ATUENR).

  The signal after noise removal is changed only on compare match between the counter and NCRJn in synchronization with the level change at the start of counting. When the coutner is stopped before the compare match, level changes at the start and ene of counting are masked and the signla after noise removal is not changed.

  When the NCEJn bit is cleared during the counter in operation, counting continues until compare match or a level change on the pin is detected.

  NCNTJ0, 1 can be read from and written to in byte or word units.

NCNTJ0, 1 is initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	NCCNTJn[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: n = 0, 1

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	NCCNTJn [7:0]	All 0	R/W	Noise Canceler Count Jn These bits store the 8-bit counter value.

13.34.11 Noise Cancel Registers J0 and J1 (NCRJ0 and NCRJ1)

NCRJ0 and NCRJ1 are 8-bit readable/writable registers that set the upper limitation of the noise canceler couters (NCNTJ1 and NCNTJ0). For example, when H'FF is set in these register, a pulse width of up to 1.64 ms is treated as noise (Pϕ = 20 MHz).

Premature-transition cancellation or minimum time-at-level cancellation is performed depending on the setting in the noise cancelltion mode register (NCMR) of the common controller. For details, see section 13.34.10, Noise Canceler Counters J0 and J1 (NCNTJ0 and NCNTJ1).

NCRJ0 and NCRJ1 can be read from and written to in byte or word units.

NCRJ0 and NCRJ1 are initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	NCTJn[7:0]							
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: n = 0, 1

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	NCTJn[7:0]	All 0	R/W	Noise Cancellation Time Jn These bits store the 8-bit TIJn noise cancellation time.

13.35 Operations of Timer J

The TCNTJn counter clock is selected by the TCRJn register from the clock-bus lines. When the TJE bit in ATUENR and the STRJn bit in TSTR is set to 1, TCNTJn can be operated. TCNTJn is cleared to H'00 by the input edge of pin TIJn. For the edge detection, the rising, falling, or both edge sensing can be selected by the IOJn bit in TCRJn.

When the FIFOEN bit in FCRFJn is set to 1, FIFOJn captures the TCNTJn counter value by the input edge of pin TIJn. The number of data captured in the FIFO is indicated in FDNRn.

When the number of data captured in FIFOJn exceeds the value in the FDFTRGJn bits in FCRJn, the FDFFJn bit in TSRJn is set to 1.

When FIFOENJn = 0, FVCRENJn = 1, and compare match between TCNTJn and OCRJn occurs, FIFOJn is enabled (FVLDFJn in TSRJn is 1). Under this conditions, when the number of data captured in FIFOJn exceeds the value in the FDFTRGJn bits in FCRJn, FIFOJn is disabled.

Operation examples for subblock 0 are shown below. The TCNTJ0 counter clock is an ideal signal to show the counting and clearing timing of TCNTJ0.

In figure 13.50, falling edge sensing is selected, FIFOEN = 1, FDFTRGJ0 = B'11, and FDFFJ0 is set by the input edge.

In figure 13.51, falling edge sensing is selected, FIFOEN = 0, FVCRENJ0 = 1, FDFTRGJ0 = B'00 and FDFF0 is set by the input edge.

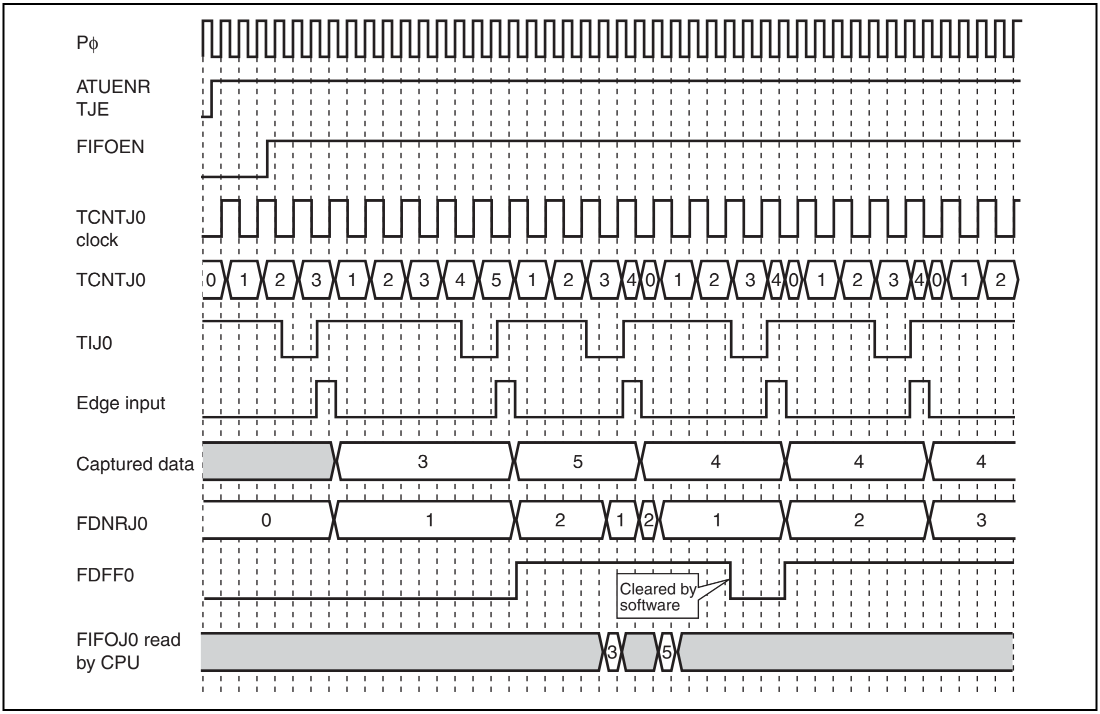


Figure 13.50 Operation Example of Timer J (1)

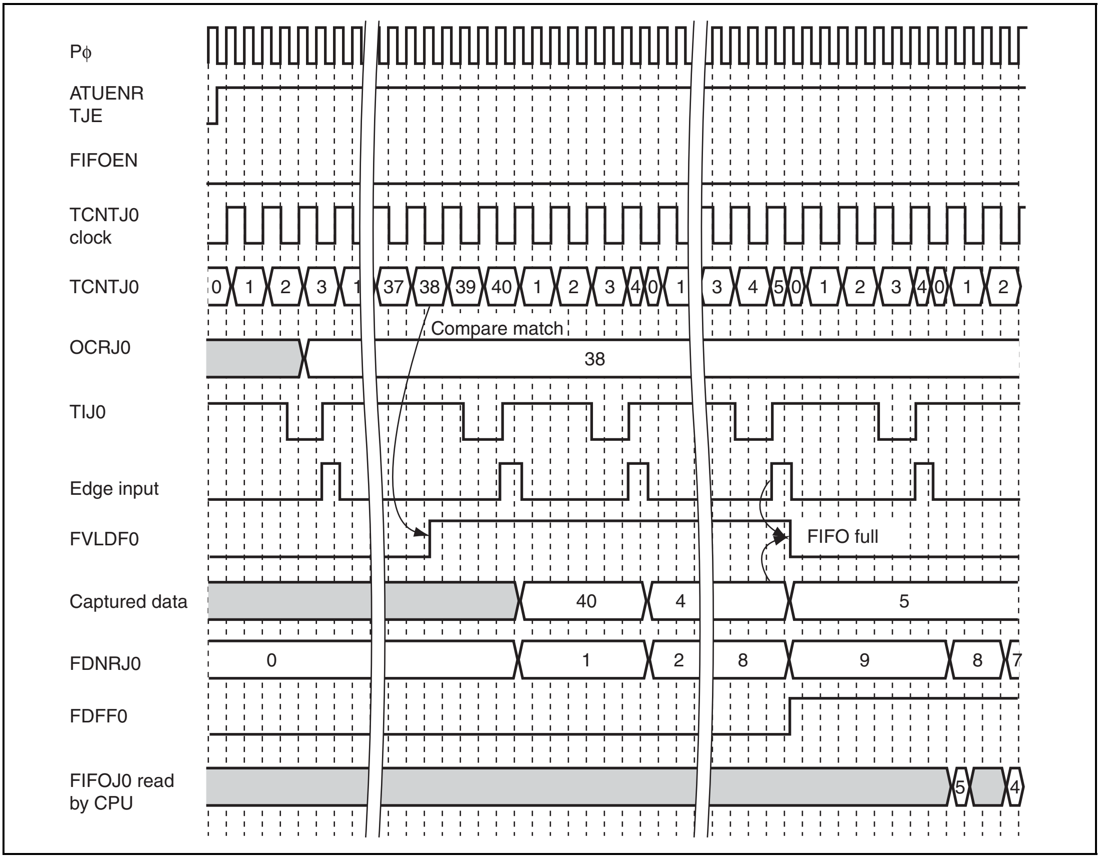


Figure 13.51 Operation Example of Timer J (2)

13.36 Usage Notes

Note that the kinds of conflicts and operation described below occur during ATU-III operation.

In this section, n denotes the subblock number and m denotes the channel number, for each timer. The values of n and m differ for each timer. For details, refer to the description of each timer.

13.36.1 Input Capture Conflict Operation

(1) Conflict between Writing to General Register and Input Capture

When a write to a general register occurs simultaneously with input capture, writing takes priority (waveforms in the left half of figure 13.52). However, if the input capture status is provided, the input capture status flag is set.

The waveforms in the right half of figure 13.52 indicate a case in which writing occurs one Pϕ cycle prior to input capture.

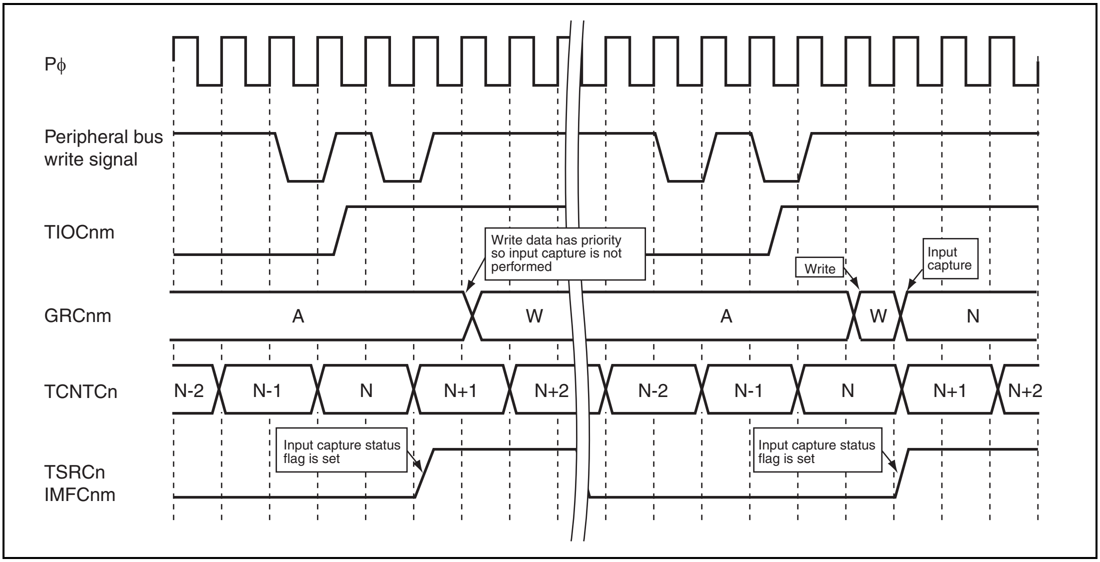

Pϕ
Peripheral bus write signal
TIOCnm
GRCnm
TCNTCn
TSRCn IMFCnm
Write data has priority so input capture is not performed
Write
Input capture
A
W
A
W
N
N-2
N-1
N
N+1
N+2
N-2
N-1
N
N+1
N+2
Input capture status flag is set
Input capture status flag is set

Figure 13.52 Conflict between Writing to GRCnm and Input Capture

Table 13.21 Resources Related to Conflict Operation between General Register Write and Input Capture

Timer	Counter (Whose Value is Captured)	Capture Register	Status
Timer C	TCNTCn	GRCnm	IMFCnm
Timer D	TCNT2Dn	GRDnm	CMFBDnm
Timer F	ECNTAFn	GRAFn	ICFFn
	ECNTBFn	GRBFn	
	ECNTCFn	GRCFn	
	ECNTCFn + GRDFn	GRDFn	

(2) Conflict between Writing to Counter and Input Capture

When a write to a counter occurs simultaneously with input capture, the value immediately before writing is captured (waveforms in the left half of figure 13.53). The waveforms in the right half of figure 13.53 indicate a case in which writing occurs one Pϕ cycle prior to input capture, so the written value is captured.

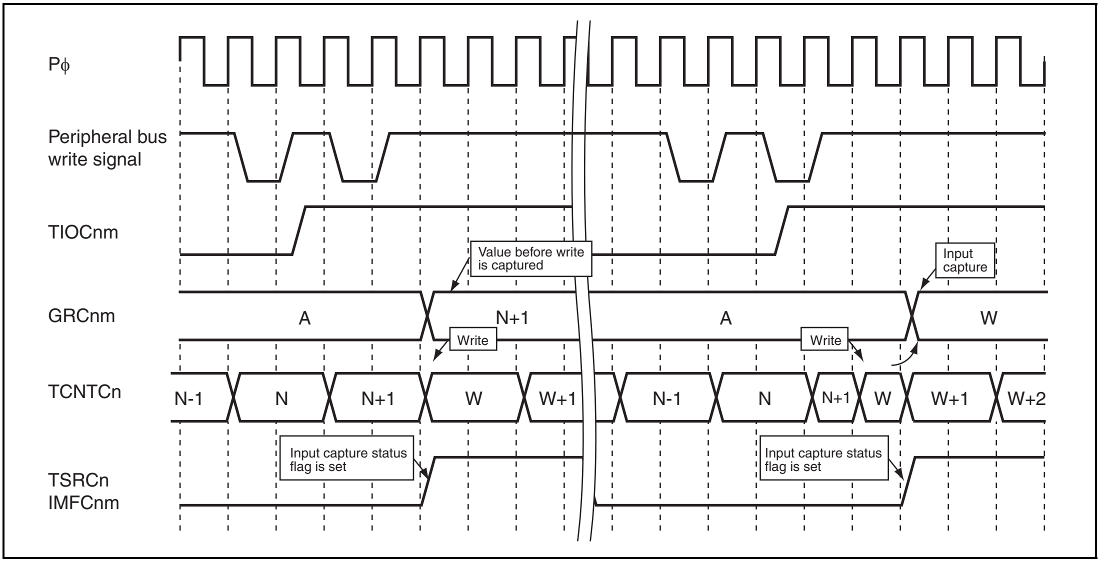

Pϕ
Peripheral bus write signal
TIOCnm
GRCnm
TCNTCn
TSRCn IMFCnm
Value before write is captured
Input capture
A
N+1
A
W
Write
Write
N-1
N
N+1
W
W+1
N-1
N
N+1
W
W+1
W+2
Input capture status flag is set
Input capture status flag is set

Figure 13.53 Conflict between Writing to TCNTCn and Input Capture

Table 13.22 Resources Related to Conflict Operation between Counter Write and Input Capture

Timer	Counter (Whose Value is Captured)	Capture Register	Status
Timer A	TCNTAn	ICRAn	ICFAn
Timer B	TCNTB0	ICRB0	ICFB0
	TCNTB0 + ICRB1	ICRB1	
Timer C	TCNTCn	GRCnm	IMFCnm
Timer D	TCNT1Dn	OSBRDn	—
	TCNT2Dn	GRDnm	—
Timer F	ECNTAFn	GRAFn	ICFFn
	ECNTBFn	GRBFn	
	ECNTCFn	GRCFn	
	ECNTCFn + GRDFn	GRDFn	
Timer J	TCNTJn	FIFOJn	FDOVFJn, FDFFJn

(3) Conflict between Setting and Clearing of Input Capture Status Flag

Flag clearing by writing 0 to it after reading it as 1 or by using the ACK signal from the DMAC takes priority over flag setting by input capture. The waveforms in the left half of figure 13.54 indicate an example in which input capture occurs simultaneously with writing 0 to the status flag to clear it, and the status flag is cleared as a result. In contrast with this, the waveforms in the right half of figure 13.54 indicate an example in which input capture is performed immediately after the flag has been cleared.

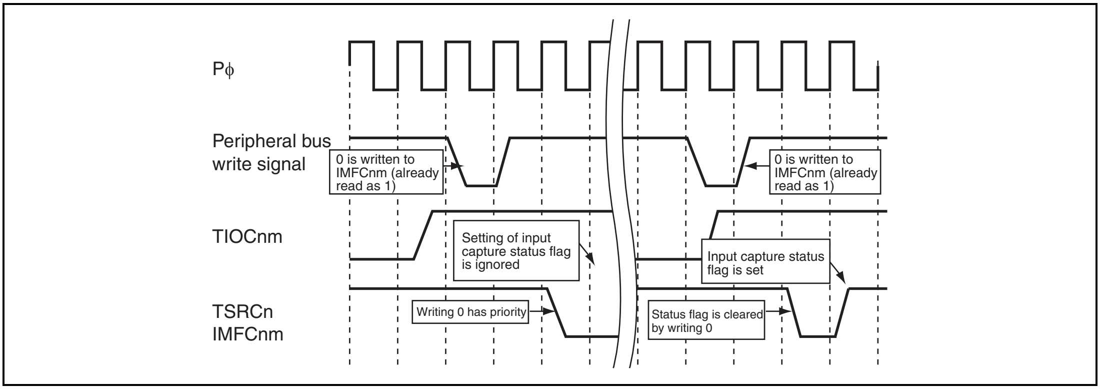

Pϕ
Peripheral bus write signal
0 is written to IMFCnm (already read as 1)
0 is written to IMFCnm (already read as 1)
TIOCnm
Setting of input capture status flag is ignored
Input capture status flag is set
TSRCn IMFCnm
Writing 0 has priority
Status flag is cleared by writing 0

Figure 13.54 Conflict between Status Flag Clearing by Writing 0 and Input Capture

Table 13.23 Resources Related to Conflict Operation between Status Flag Clearing by Writing 0 and Input Capture

Timer	Flag	Timer	Flag
Timer A	ICFAn	Timer B	ICFB0
Timer C	IMFCnm	Timer F	ICFFn
Timer J	FDOVFJn, FDFFJn	—	—

Figure 13.55 shows an example in which input capture conflicts with clearing of the status flag by the ACK signal from the DMAC.

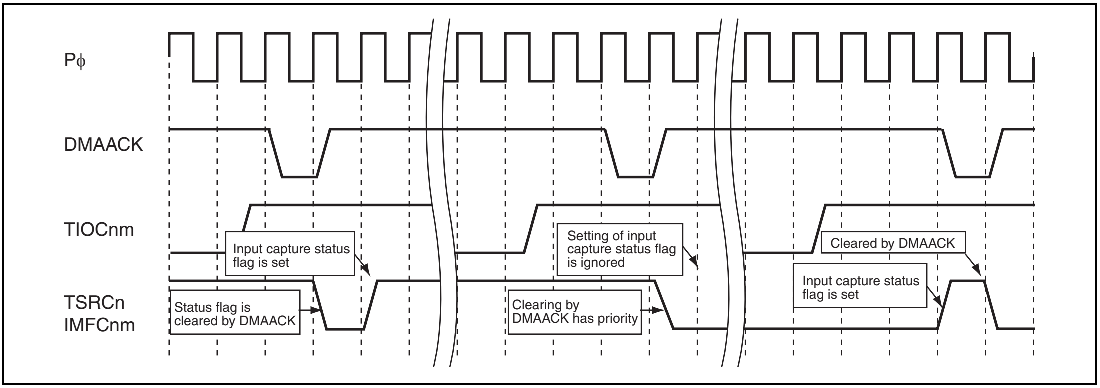

Pϕ
DMAACK
TIOCnm
TSRCn
IMFCnm
Input capture status flag is set
Status flag is cleared by DMAACK
Setting of input capture status flag is ignored
Clearing by DMAACK has priority
Cleared by DMAACK
Input capture status flag is set

Figure 13.55 Conflict between Status Flag Clearing by DMAACK and Input Capture

Table 13.24 Resources Related to Conflict Operation between Status Flag Clearing by DMAACK and Input Capture

Timer	Flag
Timer C	IMFCn0
Timer J	FDFFJn

13.36.2 Compare Match Conflict Operation

(1) Conflict between Writing to Compare-Match General Register and Compare Match

A conflict between a write to a register provided with the compare match function and compare match is described here with timer C used as an example.

If writing is performed after GRCnm and TCNTCn have matched (waveforms in the left half of figure 13.56), the compare match status flag is set. If GRCnm and TCNTCn do not match for even one Pϕ cycle (waveforms in the right half of figure 13.56), no compare match is detected.

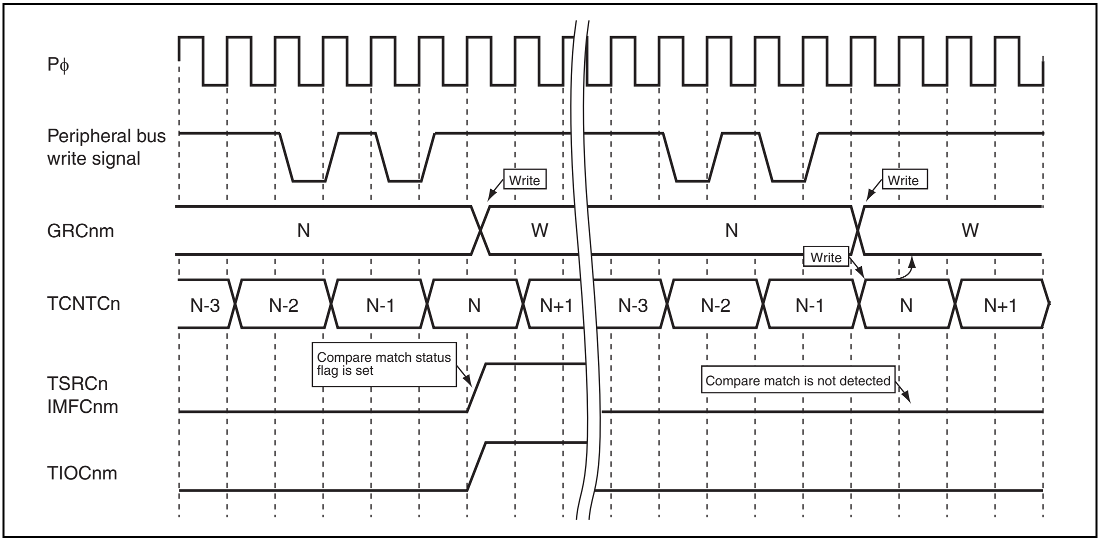

Pϕ
Peripheral bus write signal
Write
Write
GRCnm
N
W
N
W
Write
TCNTCn
N-3
N-2
N-1
N
N+1
N-3
N-2
N-1
N
N+1
Compare match status flag is set
Compare match is not detected
TSRCn IMFCnm
TIOCnm

Figure 13.56 Conflict between Writing to GRCnm and Compare Match

Table 13.25 Resources Related to Conflict Operation between Write to Compare-Match General Register and Compare Match

Timer	Counter	Compare-Match Register	Status
Timer B	TCNTB1	OCRB1	CMFB1
Timer C	TCNTCn	GRCnm	IMFCnm
Timer D	TCNT1Dn	OCRDnm	CMFADnm
	TCNT2Dn	GRDnm	CMFBDnm
Timer F	ECNTAFn	GRAFn	—
	ECNTBFn	GRBFn	—
	ECNTCFn	GRBFn	OVFCFn
Timer G	TCNTGn	OCRGn	CMFGn
Timer H	TCNT1H	OCR1H	CMFH
Timer J	TCNTJn	OCRJn	CMFJn

Note: Timing of compare match B0 and compare match B6 of timer B and the cycle match of timer E differ from the timing of these compare matches. For details, see section 13.36.2 (2), Conflict between CYLREnm Write and Cycle Match of CYLREnm–TCNTEnm.

(2) Conflict between CYLREnm Write and Cycle Match of CYLREnm–TCNTEnm

Operation when a write to CYLREnm occurs simultaneously with compare match (cycle match) with TCNTEnm is shown below. As the waveforms in the left half of figure 13.57 indicate, if CYLREnm is written to at the same time the counter is cleared by cycle match, TCNTEnm is cleared as is done on normal cycle match, and the cycle match status and PWM output also change. The waveforms in the right half of figure 13.57 show an example in which CYLREnm is written to before the counter is cleared. In this case, no cycle match is detected and TCNTEnm continues to be incremented.

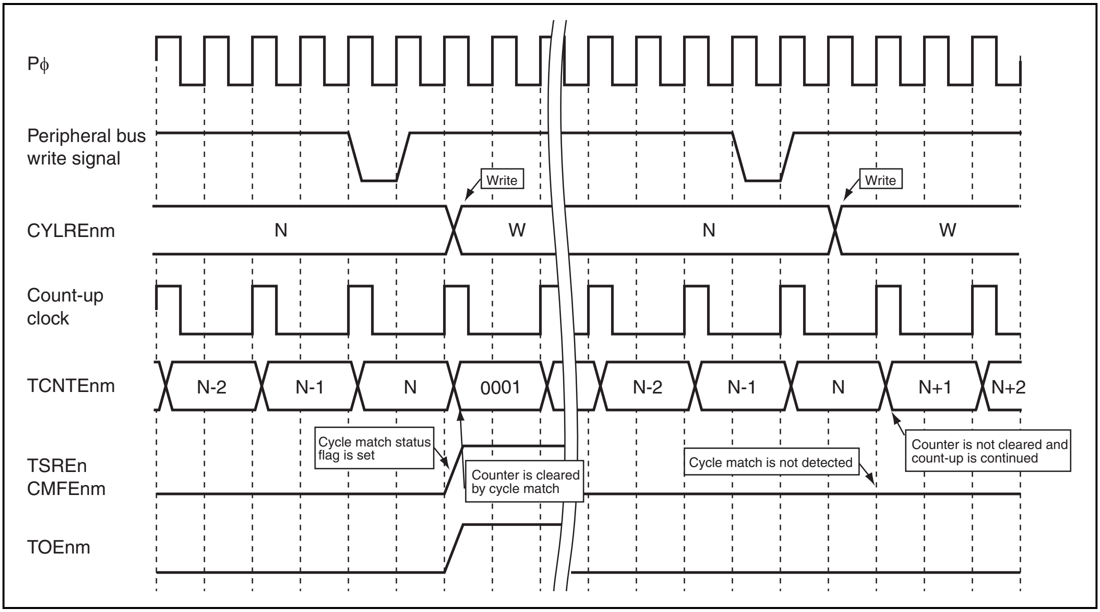

Pϕ
Peripheral bus write signal
Write
Write
CYLREnm
N
W
N
W
Count-up clock
TCNTEnm
N-2
N-1
N
0001
N-2
N-1
N
N+1
N+2
Cycle match status flag is set
Counter is cleared by cycle match
Cycle match is not detected
Counter is not cleared and count-up is continued
TSREn CMFEnm
TOEnm

Figure 13.57 Conflict between Writing to CYLREnm and Cycle Match

Table 13.26 Resources Related to Conflict Operation between Writing to Cycle Setting Register and Cycle Match with Timer Counter

Timer	Counter	Compare (Cycle) Match Register	Status
Timer B	TCNTB0	OCRB0	CMFB0
	TCNTB6	OCRB6/OCRB7	CMFB6
Timer E	TCNTEnm	CYLREnm	CMFEnm

(3) Conflict between Writing to Counter and Compare Match

A conflict between a write to a counter and compare match is described below. If writing is performed after the compare match register and counter values have matched (waveforms in the left half of figure 13.58), the compare match status flag is set. If the compare match register and counter do not match for even one Pϕ cycle (waveforms in the right half of figure 13.58), no compare match is detected.

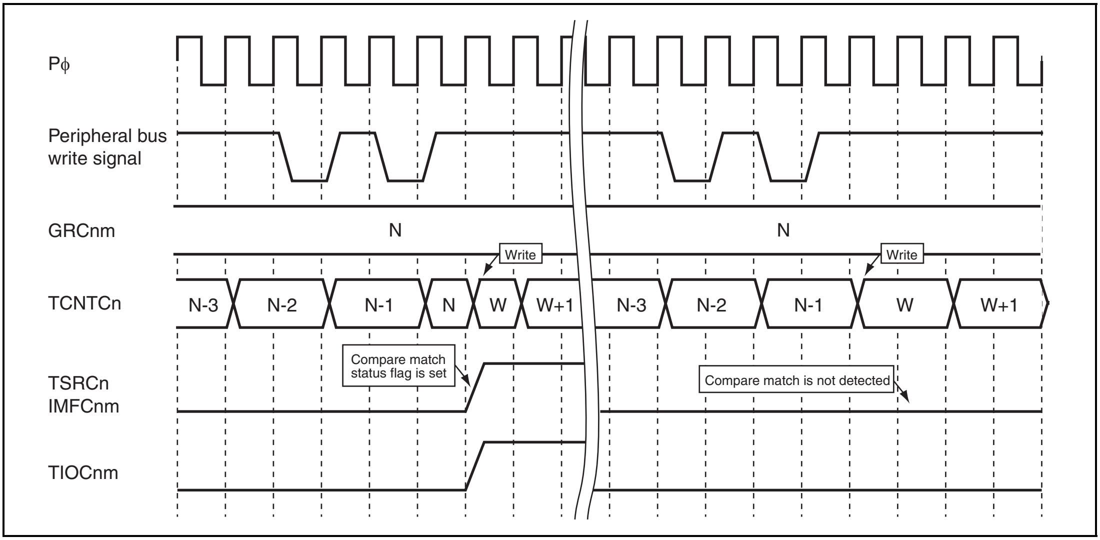

Pϕ
Peripheral bus write signal
GRCnm
N
N
Write
Write
TCNTCn
N-3
N-2
N-1
N
W
W+1
N-3
N-2
N-1
W
W+1
Compare match status flag is set
Compare match is not detected
TSRCn IMFCnm
TIOCnm

Figure 13.58 Conflict between Writing to TCNTCn and Compare Match

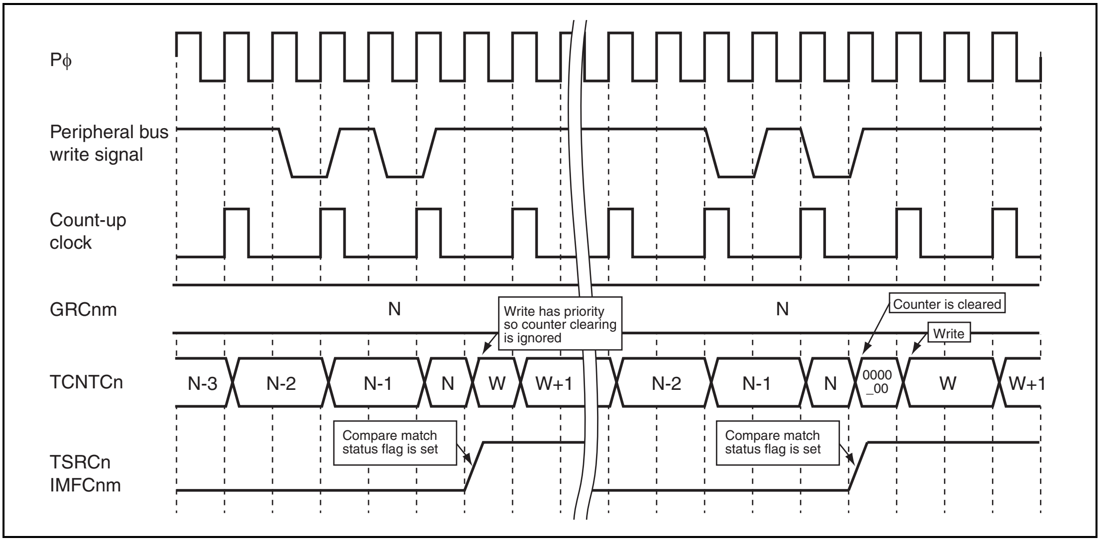
Table 13.27 Resources Related to Conflict Operation between Writing to Counter and Compare Match

Timer	Counter	Compare Match Register	Status
Timer B	TCNTB1	OCRB1	CMFB1
Timer C	TCNTCn	GRCnm	IMFCnm
Timer D	TCNT1Dn	OCRDnm	CMFADnm
	TCNT2Dn	GRDnm	CMFBDnm
Timer F	ECNTAFn	GRAFn	—
	ECNTBFn	GRBFn	—
	ECNTCFn	GRBFn	OVFCFn
Timer G	TCNTGn	OCRGn	CMFGn
Timer H	TCNT1H	OCR1H	CMFH
Timer J	TCNTJn	OCRJn	CMFJn

Note: Timing of compare match B0 and compare match B6 of timer B and the cycle match of timer E differ from the timing of these compare matches. For details, see section 13.36.2 (5), Conflict between Writing to TCNTEnm and Counter Clearing by Cycle Match.

(4) Conflict between Writing to Counter and Counter Clearing by Compare Match

The waveforms shown here are for when the function to clear a counter by compare match is enabled. When a write to a counter occurs simultaneously with counter clearing by compare match, the counter is not cleared and writing takes priority (waveforms in the left half of figure 13.59). However, the compare match status flag is set. The waveforms in the right half of figure 13.59 show a case in which writing to TCNTCn is one Pϕ cycle later.



Pϕ
Peripheral bus write signal
Count-up clock
GRCnm
N
N
Write has priority so counter clearing is ignored
Counter is cleared
Write
TCNTCn
N-3
N-2
N-1
N
W
W+1
N-2
N-1
N
0000 _00
W
W+1
Compare match status flag is set
Compare match status flag is set
TSRCn IMFCnm

Figure 13.59 Conflict between Writing to TCNTCn and Counter Clearing by Compare Match

Table 13.28 Resources Related to Conflict Operation between Writing to Counter and Counter Clearing by Compare Match

Timer	Counter	Compare Match Register	Status
Timer C	TCNTCn	GRCnm	IMFCnm
Timer F	ECNTAFn	GRAFn	—
	ECNTBFn	GRBFn	—
Timer G	TCNTGn	OCRGn	CMFGn
Timer H	TCNT1H	OCR1H	CMFH

(5) Conflict between Writing to TCNTEnm and Counter Clearing by Cycle Match

When a write to TCNTEnm occurs simultaneously with counter clearing by cycle match, the counter is not cleared and TCNTEnm is written to. Note that the cycle match status flag is set, and the cycle setting register and duty cycle setting register are reloaded (waveforms in the right half of figure 13.60). In addition, the PWM waveforms are output as they are in cycle match.

The waveforms in the left half of figure 13.60 show operation when writing is performed one Pϕ cycle earlier than the count-up clock.

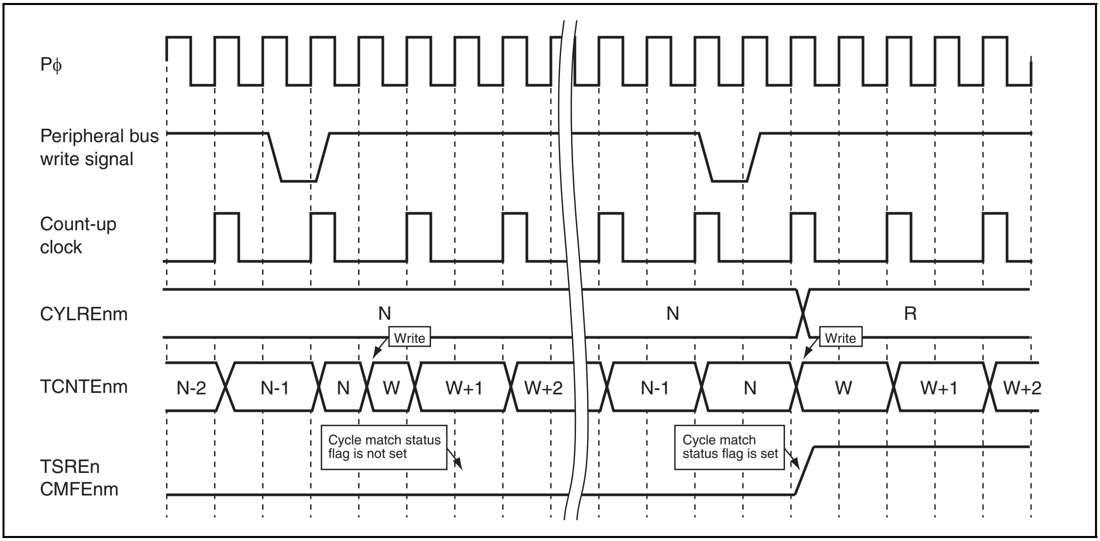

Pϕ
Peripheral bus write signal
Count-up clock
CYLREnm
N
N
R
Write
Write
TCNTEnm
N-2
N-1
N
W
W+1
W+2
N-1
N
W
W+1
W+2
Cycle match status flag is not set
Cycle match status flag is set
TSREn CMFEnm

Figure 13.60 Conflict between TCNTEnm Write and Counter Clearing by Cycle Match

Table 13.29 Resources Related to Conflict Operation between Counter Write and Counter Clearing by Cycle Match

Timer	Counter	Compare (Cycle) Match Register	Status
Timer B	TCNTB0	OCRB0	CMFB0
	TCNTB6	OCRB6/OCRB7	CMFB6
Timer E	TCNTEnm	CYLREnm	CMFEnm

(6) Conflict between Setting and Clearing of Compare Match Status Flag

Flag clearing by writing 0 to the flag after reading it as 1 takes priority over flag setting by compare match (waveforms in the left half of figure 13.61). The waveforms in the right half of figure 13.61 show the way compare match occurs immediately after the status flag has been cleared by writing 0 to it.

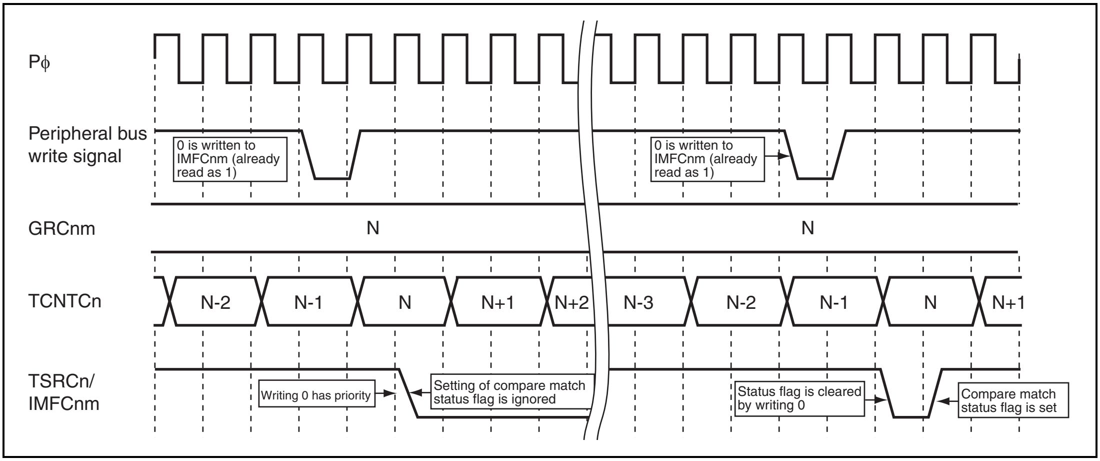

Pϕ
Peripheral bus write signal
0 is written to IMFCnm (already read as 1)
0 is written to IMFCnm (already read as 1)
GRCnm
N
N
TCNTCn
N-2
N-1
N
N+1
N+2
N-3
N-2
N-1
N
N+1
TSRCn/ IMFCnm
Writing 0 has priority
Setting of compare match status flag is ignored
Status flag is cleared by writing 0
Compare match status flag is set

Figure 13.61 Conflict between Setting and Clearing of Compare Match Status Flag

Table 13.30 Resources Related to Conflict Operation between Setting and Clearing of Compare Match Status Flag

Timer	Counter	Compare Match Register	Status
Timer B	TCNTB1	OCRB1	CMFB1
Timer C	TCNTCn	GRCnm	IMFCnm
Timer D	TCNT1Dn	OCRDnm	CMFADnm
	TCNT2Dn	GRDnm	CMFBDnm
Timer G	TCNTGn	OCRGn	CMFGn
Timer H	TCNT1H	OCR1H	CMFH
Timer J	TCNTJn	OCRJn	CMFJn

Note: Timing of compare match B0 and compare match B6 of timer B and the cycle match of timer E differ from the timing of these compare matches. For details, see section 13.36.2 (8), Conflict between Setting of Cycle Match Status Flag and Clearing by Writing 0.

(7) Conflict between Setting of Compare Match Status Flag and Clearing by DMAACK

When setting of the compare match status flag occurs simultaneously with the DMAACK signal, clearing of the status flag due to the DMAACK signal takes priority.

Table 13.31 Resource Related to Conflict Operation between Setting of Compare Match Status Flag and Clearing by DMAACK

Timer	Counter	Compare Match Register	Status
Timer G	TCNTGn	OCRGn	CMFGn

(8) Conflict between Setting of Cycle Match Status Flag and Clearing by Writing 0

When setting of the cycle match status flag (cycle match) occurs simultaneously with writing 0 to the status flag after reading it as 1, writing 0 takes priority. The waveforms in the left half of figure 13.62 indicate an example in which setting of the flag due to cycle match occurs simultaneously with clearing of the flag by 0 written to it. The waveforms in the right half of figure 13.62 show an example in which the flag is cleared one Pϕ cycle earlier than flag setting.

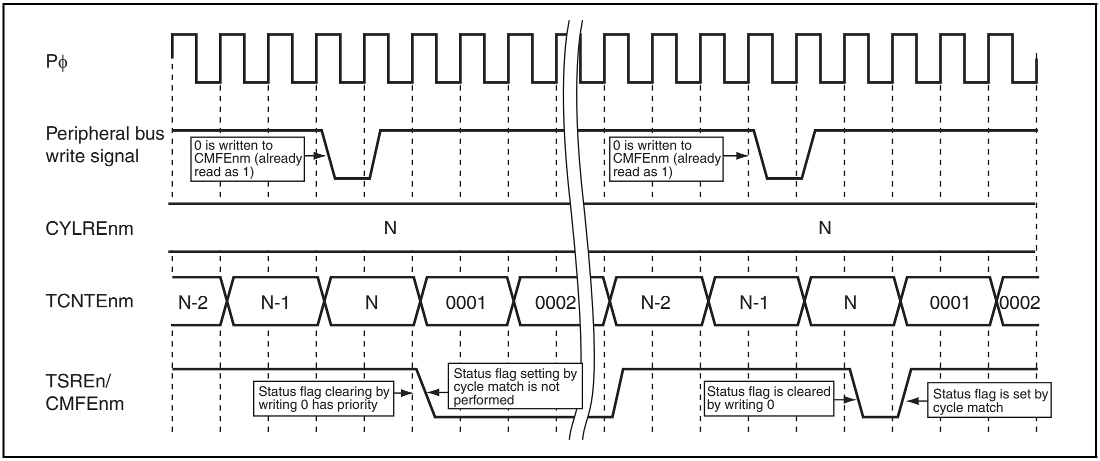

Pϕ
Peripheral bus write signal
0 is written to CMFEnm (already read as 1)
0 is written to CMFEnm (already read as 1)
CYLREnm
N
N
TCNTEnm
N-2
N-1
N
0001
0002
N-2
N-1
N
0001
0002
TSREn/ CMFEnm
Status flag clearing by writing 0 has priority
Status flag setting by cycle match is not performed
Status flag is cleared by writing 0
Status flag is set by cycle match

Figure 13.62 Conflict between Clearing of Cycle Match Status Flag by Writing 0 and Cycle Match

Table 13.32 Resources Related to Conflict Operation between Setting of Cycle Match Status Flag and Clearing by Writing 0

Timer	Counter	Compare (Cycle) Match Register	Status
Timer B	TCNTB0	OCRB0	CMFB0
	TCNTB6	OCRB6	CMFB6
Timer E	TCNTEnm	CYLREnm	CMFEnm

(9) Conflict between Setting of Cycle Match Status Flag and Clearing by DMAACK

When setting of the cycle match status flag (cycle match) occurs simultaneously with the DMAACK signal, clearing of the status flag due to the DMAACK signal takes priority.

Table 13.33 Resource Related to Conflict Operation between Setting of Cycle Match Status Flag and Clearing by DMAACK

Timer	Counter	Cycle Match Register	Status
Timer E	TCNTEnm	CYLREnm	CMFEnm

(10) Conflict between Status Flag Setting by Forced Compare Match and Status Flag Clearing by DMAACK

When the status flag is set by forced compare match at the same time the status flag is cleared by the DMAACK signal, clearing of the status flag takes priority (waveforms in the left half of figure 13.63). On the other hand, the waveforms in the right half of figure 13.63 show an example in which the DMAACK signal is input to clear the status flag during the cycle in which 1 is written to the forced compare match bit. In this case, the attempt to write 1 to the forced compare match bit is ignored and it is cleared together with the status flag.

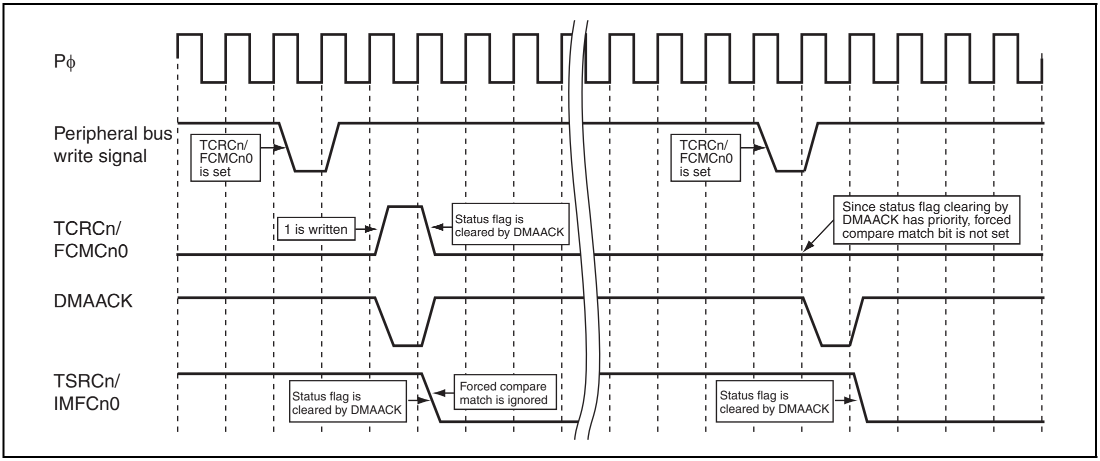

Pϕ
Peripheral bus write signal
TCRCn/ FCMCn0 is set
TCRCn/ FCMCn0 is set
TCRCn/ FCMCn0
1 is written
Status flag is cleared by DMAACK
Since status flag clearing by DMAACK has priority, forced compare match bit is not set
DMAACK
TSRCn/ IMFCn0
Status flag is cleared by DMAACK
Forced compare match is ignored
Status flag is cleared by DMAACK

Figure 13.63 Conflict between Forced Compare Match and Status Flag Clearing by DMAACK

Table 13.34 Resource Related to Conflict Operation between Status Flag Setting by Forced Compare Match and Status Flag Clearing by DMAACK

Timer	Counter	Compare Match Register	Status
Timer C	TCNTCn	GRCnm	IMFCnm

(11) Conflict between Detection of 1H Compare Match and Disabling of Counter by ATUENR Setting

When compare match between TCNT1H and OCR1H is detected at the same time a counter enable bit (each bit in ATUENR) is changed to 0, compare match is detected but TCNT2H is not incremented (waveforms in the left half of figure 13.64). Thereafter, even though a counter enable bit is set to 1, TCNT2H is not incremented until the next compare match occurs.

The waveforms in the right half of figure 13.64 show an example in which a counter enable bit is changed to 0 one Pϕ cycle after compare match has been detected.

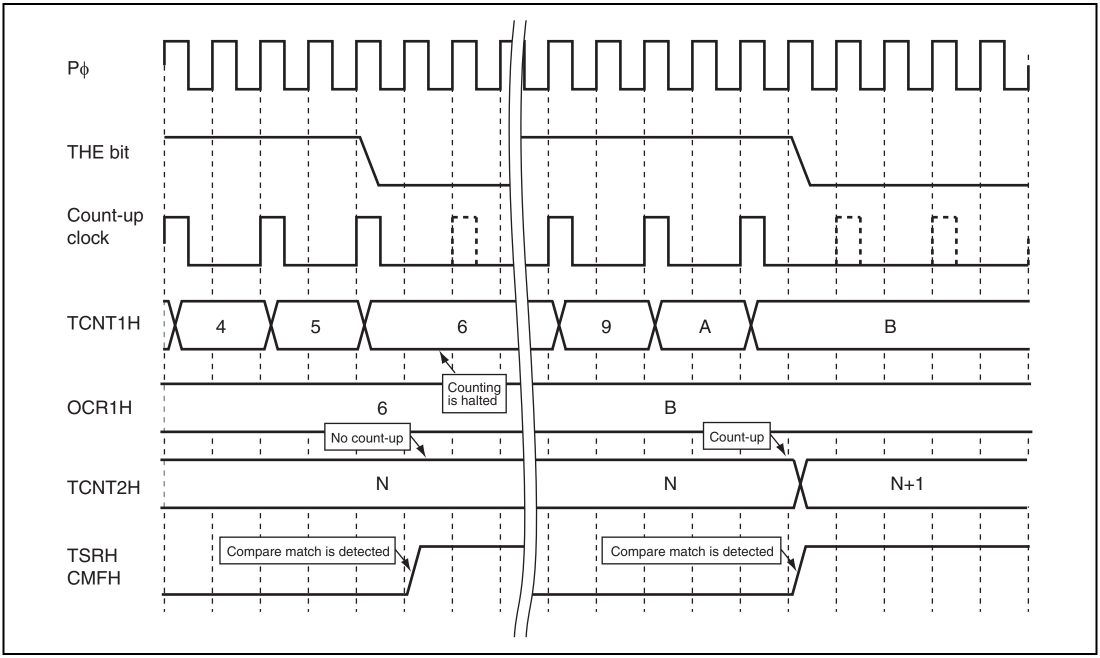


Figure 13.64 Conflict between Compare Match Detection and Disabling of Counter

Table 13.35 Resource Related to Conflict Operation between Compare Match Detection and Clearing of Counter Enable Bit

Timer	Counter	Compare Match Register	Status
Timer H	TCNT1H	OCR1H	CMFH

(12) Conflict between Writing 0 to TCNTEnm and Cycle Match

Operation when writing 0 to TCNTEnm occurs simultaneously with cycle match is shown below. The waveforms in the left half of figure 13.65 show a case in which H'0000 is written to TCNTEnm at the same time TCNTEnm is to be cleared to H'0001 due to cycle match. Though the cycle match status flag is set, PWM output is not started because writing 0 takes priority. PWM output is restarted when TCNTEnm is incremented to H'0001.

The waveforms in the middle of figure 13.65 indicate a case in which 0 is written to TCNTEnm one Pϕ cycle after the counter has been cleared by cycle match. Cycle match detection and PWM output are restarted at the timing the TCNTEnm counter value changes from N to 1. In contrast with this, the waveforms in the right half of figure 13.65 show an example in which 0 is written one Pϕ cycle before detection of cycle match. In this case, neither cycle match is detected nor PWM output restarted, and the previous state is retained.

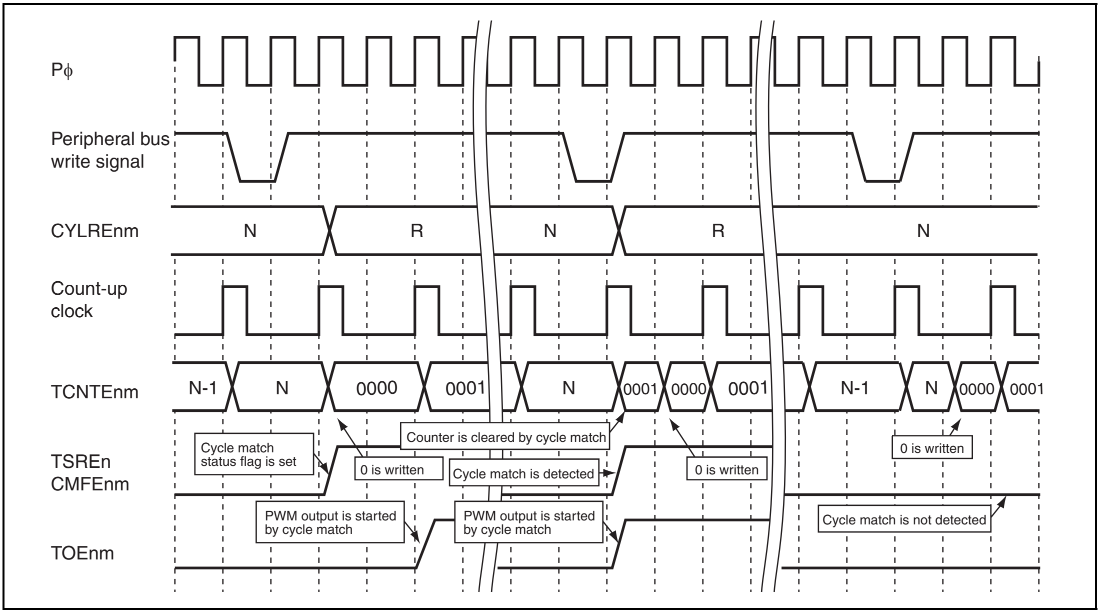

Pϕ
Peripheral bus write signal
CYLREnm
N
R
N
R
N
Count-up clock
TCNTEnm
N-1
N
0000
0001
N
0001
0000
0001
N-1
N
0000
0001
TSREn CMFEnm
TOEnm
Cycle match status flag is set
0 is written
Counter is cleared by cycle match
Cycle match is detected
0 is written
0 is written
PWM output is started by cycle match
PWM output is started by cycle match
Cycle match is not detected

Figure 13.65 Conflict between Writing 0 to TCNTEnm and Cycle Match

13.36.3 Load/Reload Conflict Operation

(1) Conflict between Data Transfer and Writing to Transfer Destination Register

A conflict between data transfer between registers and a peripheral bus write to the transfer destination register is described below.

When data transfer occurs simultaneously with a write to the transfer destination register, writing takes priority and the attempt of data transfer is ignored. Figure 13.66 shows a conflict between reload to CYLREnm of timer E and a write to it.

As shown by the waveforms in the left half of figure 13.66, if writing to CYLREnm occurs at the same timing as cycle reload, writing takes priority. The waveforms in the right half of figure 13.66 indicate a case in which CYLREnm is written to immediately after cycle reload.

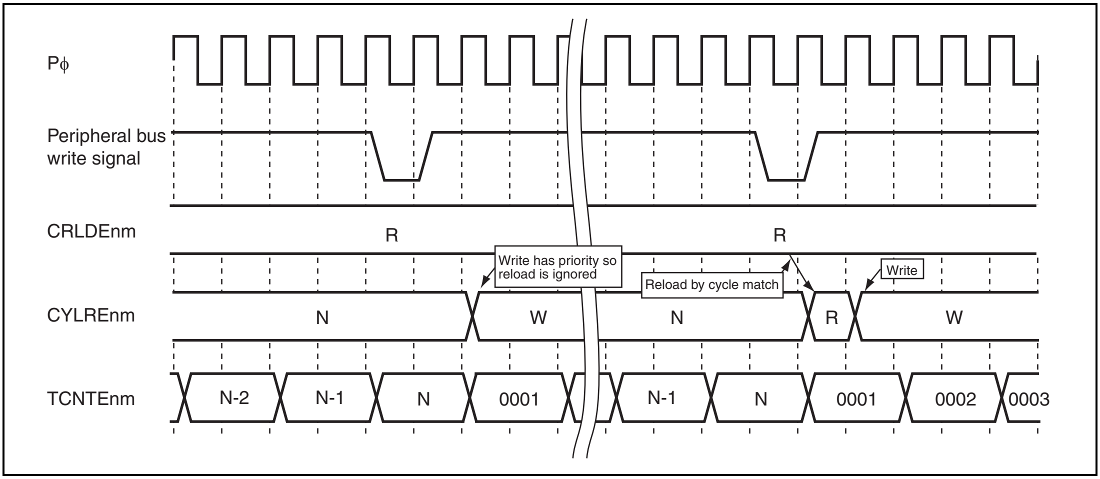


Figure 13.66 Conflict between Writing to CYLREnm and Cycle Reload

Table 13.36 Resources Related to Conflict Operation between Data Transfer and Writing to Transfer Destination Register

Timer	Transfer Data	Transfer Destination Register	Transfer Timing
Timer B	ICRB0 LDB TCNTB2 – PIMR TCNTB2 + RLDB	TCNTB2	External event
	ICRB0 – PIMR LDB – PIMR	RLDB	External event
	TCNTB3 + PIMR	TCNTB3	External event
	TCNTB3	TCNTB4	External event
Timer E	CRLDEnm	CYLREnm	Cycle match
	DRLDEnm	DTREnm	Cycle match

(2) Conflict between Data Transfer and Writing to Transfer Source Register

A conflict between data transfer between registers and a peripheral bus write to the transfer source register is described below. When data transfer occurs simultaneously with a write to the transfer source register, the value prior to writing is transferred. At the same time, the value of the transfer source register is modified. Operation when writing to CRLDEnm occurs at the timing of cycle reload is shown below. If writing to CRLDEnm occurs at the same timing as cycle reload (waveforms in the left half of figure 13.67), the value immediately before writing is reloaded. On the other hand, the waveforms in the right half of figure 13.67 show an example in which CRLDEnm is written to one cycle earlier than cycle reload.

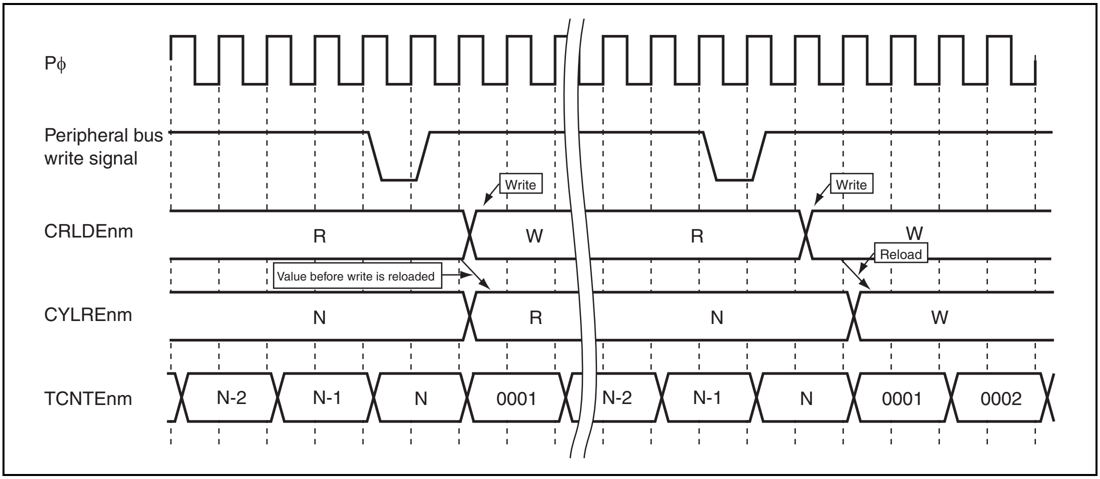


Figure 13.67 Conflict between Writing to CRLDEnm and Cycle Reload

Table 13.37 Resources Related to Conflict Operation between Data Transfer and Writing to Transfer Source Register

Timer	Transfer Source Register	Transferred Value	Transfer Destination Register	Transfer Timing
Timer B	LDB	LDB – PIMR	RLDB	External event
		LDB	TCNTB2	External event
	PIMR	ICRB0 – PIMR LDB – PIMR	RLDB	External event
		TCNTB2 – PIMR	TCNTB2	External event
		TCNTB3 + PIMR	TCNTB3	External event
	RLDB	TCNTB2 + RLDB	TCNTB2	External event
	TCNTB3	TCNTB3	TCNTB4	External event
Timer E	CRLDEnm	CRLDEnm	CYLREnm	Cycle match
	DRLDEnm	DRLDEnm	DTREnm	Cycle match

13.36.4 Counter Conflict Operation

(1) Conflict between Writing to Counter and Count-Up/Count-Down

When a write to a counter occurs simultaneously with incrementation/decrementation of the counter, the write operation takes priority. The attempt to increment/decrement the value is ignored and incrementation/decrementation recommences from the new value on the next counter clock.

(2) Conflict between Count-Up and Counter Clearing

When incrementation of a counter occurs simultaneously with clearing of the counter, the counter is not cleared to 0 but cleared to 1.

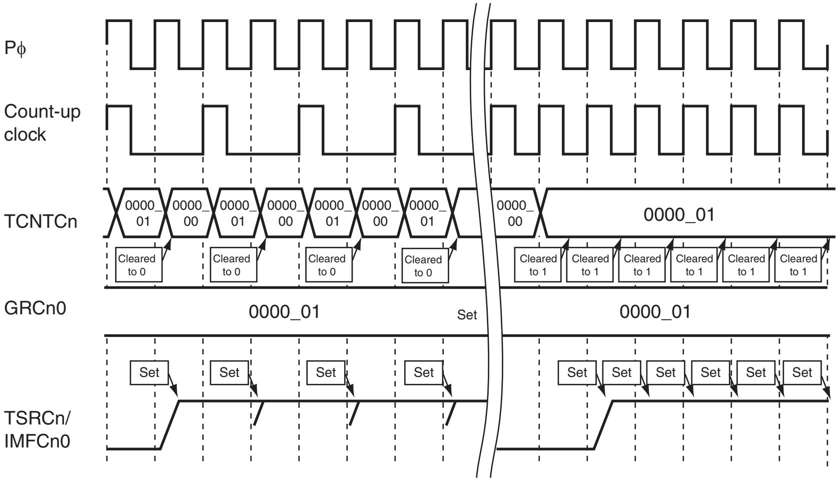


Figure 13.68 Simultaneous Occurrence of Count-Up and Counter Clearing

Table 13.38 Resources Related to Conflict Operation between Count-Up and Counter Clearing

Timer	Counter	Compare Match Register	Remarks
Timer C	TCNTCn	GRCn0	Only when PWMn0 = 1
Timer F	ECNTAFn	GRAFn	Only when MDFn = 000, 110, or 111
	ECNTBFn	GRBFn	Only when MDFn = 001
Timer G	TCNTGn	OCRGn	
Timer H	TCNT1H	OCR1H	

(3) Conflict between Writing to Counter and Overflow

When counter overflow occurs simultaneously with a write to TCNTCn, writing to TCNTCn takes priority. However, the overflow status flag is set (waveforms in the left half of figure 13.69). If the timing for writing to the counter is earlier than incrementation of the counter (waveforms in the right half of figure 13.69), the overflow status flag is not set.

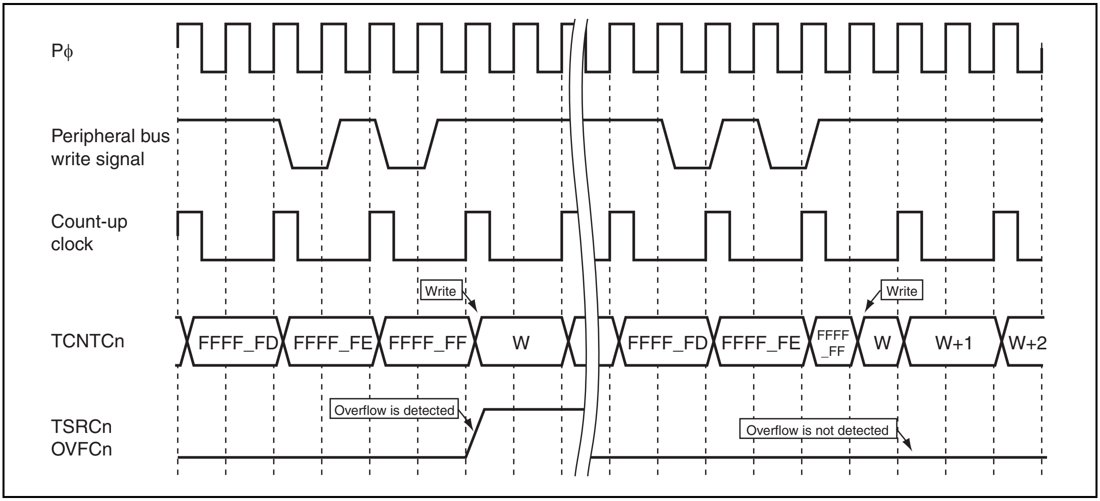


Figure 13.69 Conflict between Writing to TCNTCn and Counter Clearing on Overflow

Table 13.39 Resources Related to Conflict Operation between Writing to Counter and Overflow

Timer	Counter	Status
Timer A	TCNTA	OVFA
Timer C	TCNTCn	OVFCn
Timer D	TCNT1Dn	OVF1Dn
	TCNT2Dn	OVF2Dn
Timer E	TCNTEnm	OVFEnm
Timer F	ECNTAFn	OVFAFn
	ECNTBFn	OVFBFn
	ECNTCFn	OVFCFn
Timer G	TCNTGn	OVFGn
Timer H	TCNT1H	OVF1H
	TCNT2H	OVF2H
Timer J	TCNTJn	OVFJn

(4) Conflict between Setting and Clearing of Overflow Status Flag

When clearing and setting of the overflow status flag occur simultaneously, clearing takes priority. Shown below is an example in which the status flag is set by the counter value overflowing from H'FFFF FF to H'0000 00 at the same time the status flag is cleared by 0 written to it (waveforms in the left half of figure 13.70). The waveforms in the right half of figure 13.70 show the way the overflow status flag is set again immediately after the status flag has been cleared.

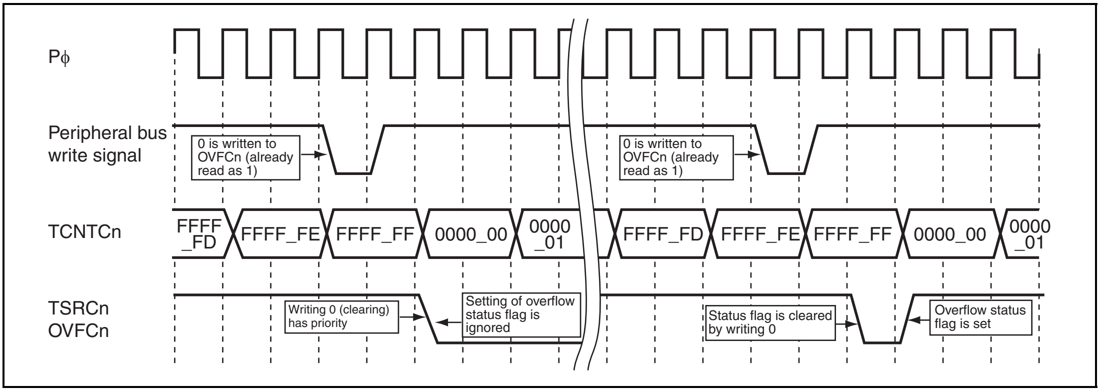

Pϕ
Peripheral bus write signal
0 is written to OVFCn (already read as 1)
0 is written to OVFCn (already read as 1)
TCNTCn
FFFF_FD
FFFF_FE
FFFF_FF
0000_00
0000_01
FFFF_FD
FFFF_FE
FFFF_FF
0000_00
0000_01
TSRCn OVFCn
Writing 0 (clearing) has priority
Setting of overflow status flag is ignored
Status flag is cleared by writing 0
Overflow status flag is set

Figure 13.70 Conflict between Setting and Clearing of Overflow Status Flag

Table 13.40 Resources Related to Conflict Operation between Setting and Clearing of Overflow Status Flag

Timer	Counter	Status
Timer A	TCNTA	OVFA
Timer C	TCNTCn	OVFCn
Timer D	TCNT1Dn	OVF1Dn
	TCNT2Dn	OVF2Dn
Timer E	TCNTEnm	OVFEnm
Timer F	ECNTAFn	OVFAFn
	ECNTBFn	OVFBFn
	ECNTCFn	OVFCFn
Timer G	TCNTGn	OVFGn
Timer H	TCNT1H	OVF1H
	TCNT2H	OVF2H
Timer J	TCNTJn	OVFJn

(5) Conflict between Overflow and Counter Clearing by Compare Match

If the maximum value is set in a compare match register that has the function to clear a counter by compare match and that function is enabled, when the counter reaches its maximum value, the counter is cleared. No overflow is detected even if the count-up clock frequency is the same as Pϕ.

Examples using TCNT1H and OCR1H of timer H are shown in figure 13.71. With H'FFFF set in OCR1H, a case in which the count-up clock is not Pϕ × 1/1 (waveforms in the left half of figure 13.71) and a case in which the count-up clock is Pϕ × 1/1 (waveforms in the right half of figure 13.71) are shown.

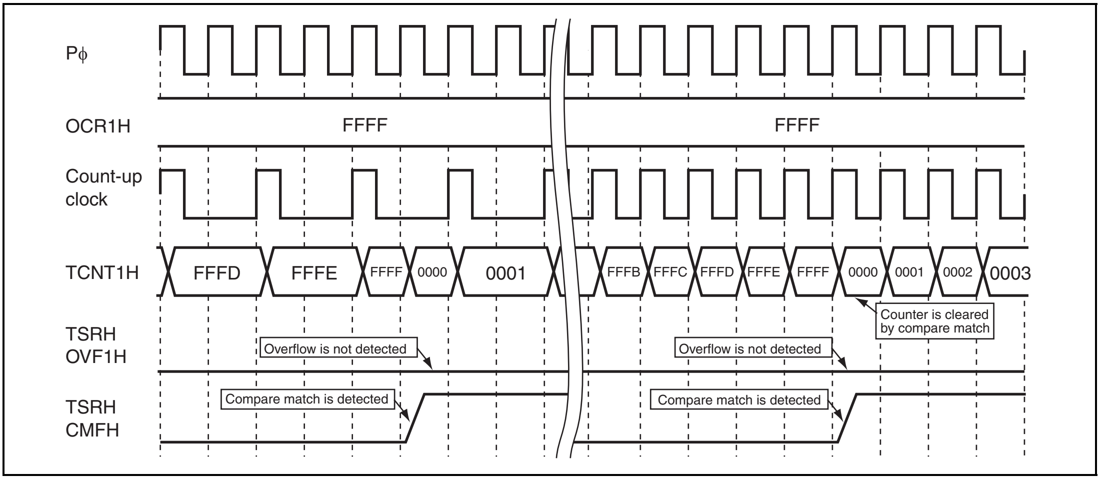


Figure 13.71 Conflict between TCNT1H Counter Overflow and Compare Match

Table 13.41 Resources Related to Conflict Operation between Counter Overflow and Compare Match

Timer	Counter	Status	Remarks
Timer C	TCNTCn	OVFCn	Only when PWMn0 = 1
Timer F	ECNTAFn	OVFAFn	Only when MDFn = 000, 110, or 111
	ECNTBFn	OVFBFn	Only when MDFn = 001
Timer G	TCNTGn	OVFGn	
Timer H	TCNT1H	OVF1H	

When the function to clear a counter by compare match is not provided or when that function is disabled, the overflow status flag is set. Figure 13.72 shows operation when the PWMn0 bit of timer C is 1 (counter clearing is enabled) and when the PWMn0 bit is 0 (counter clearing is disabled).

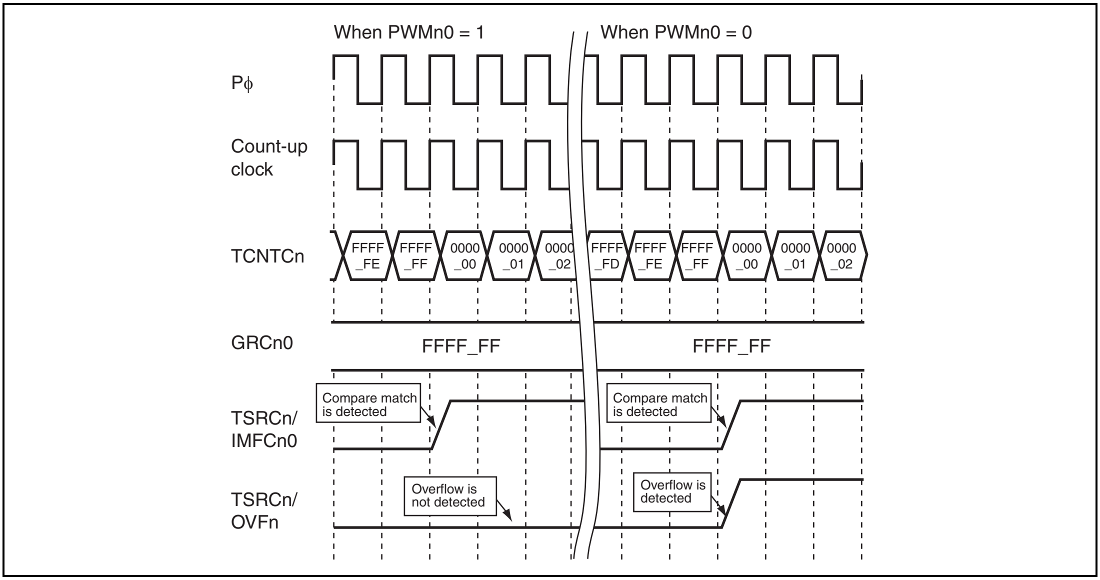


Figure 13.72 Conflict between Counter Clearing by Compare Match of Timer C and Overflow (PWMn0 = 1/0)

13.36.5 Noise Canceler Conflict Operation

Conflicts in the noise cancelers are described here.

Table 13.42 Resources Related to Conflict Operation between Writing to Noise Canceler Counter and Compare Match with Noise Canceler Register

Timer	Counter	Compare Match Register
Timer A	NCNTAn	NCRAn
Timer C	NCNTCnm	NCRCnm
Timer F	NCNTAFn	NCRAFn
	NCNTBFn	NCRBFn
Timer J	NCNTJn	NCRJn

(1) Conflict between Writing to Noise Canceler Counter and Compare Match with Noise Canceler Register

When a write to NCNT occurs simultaneously with a compare match with NCR, writing takes priority. An example in minimum time-at-level cancellation mode is shown below. In the example in the left half of figure 13.73, since writing prevents compare match from occurring, input capture is also not performed. The example in the right half of figure 13.73 shows a case in which writing is performed one Pϕ cycle later. In this case, compare match occurs so input capture processing is carried out.

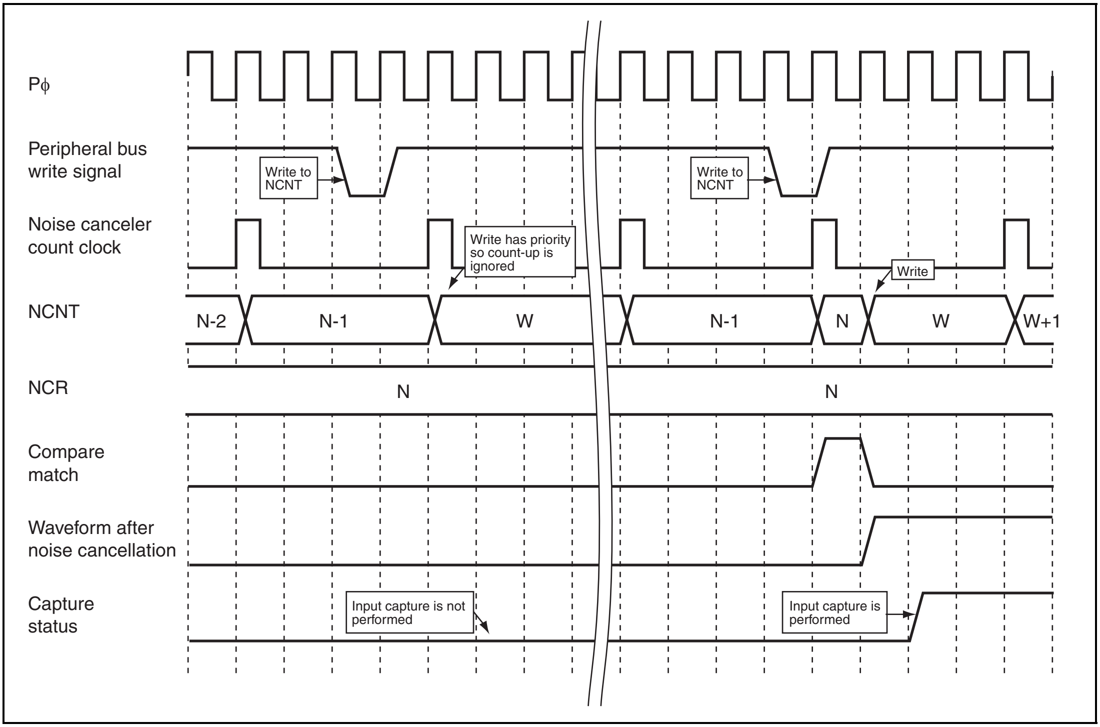

Pϕ
Peripheral bus write signal
Write to NCNT
Write to NCNT
Noise canceler count clock
Write has priority so count-up is ignored
Write
NCNT
N-2
N-1
W
N-1
N
W
W+1
NCR
N
N
Compare match
Waveform after noise cancellation
Capture status
Input capture is not performed
Input capture is performed

Figure 13.73 Conflict between Writing to NCNT and Compare Match of NCNT–NCR (Example in Minimum Time-At-Level Cancellation Mode)

(2) Conflict between Writing to Noise Canceler Register and Compare Match with Noise Canceler Counter

When a write to NCR occurs simultaneously with a compare match with NCNT, writing takes priority. An example in minimum time-at-level cancellation mode is shown below. In the example in the left half of figure 13.74, since writing prevents compare match from occurring, input capture is also not performed. The example in the right half of figure 13.74 shows a case in which writing is performed one Pϕ cycle later. In this case, compare match occurs so input capture processing is carried out.

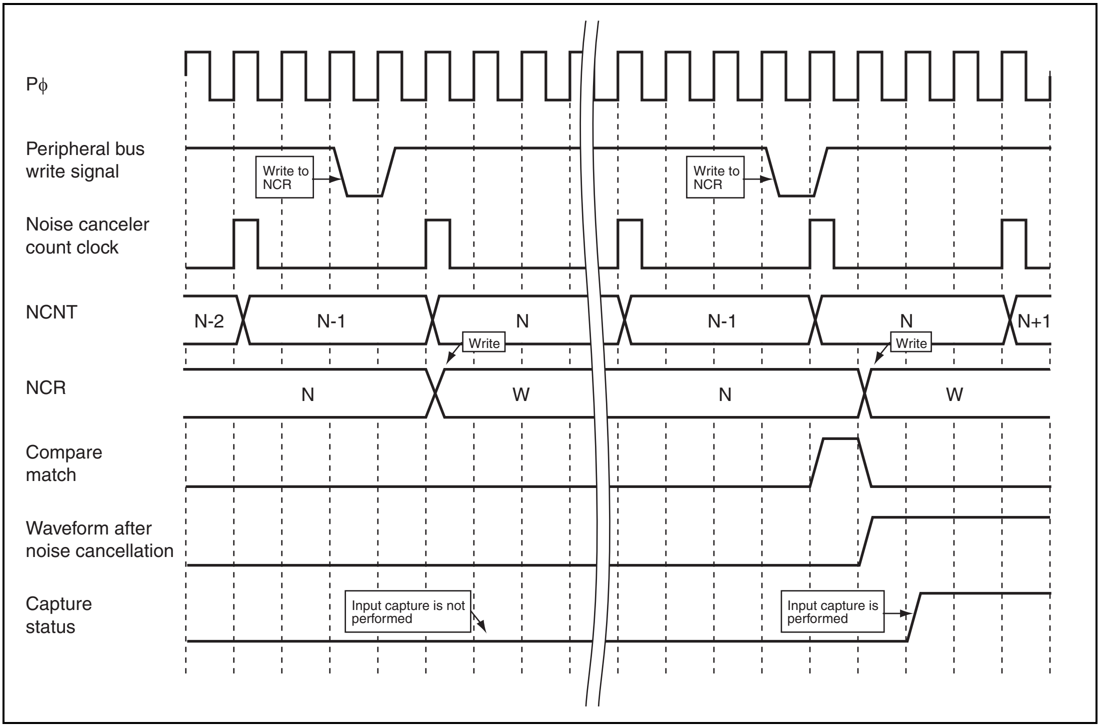

Pϕ
Peripheral bus write signal
Write to NCR
Write to NCR
Noise canceler count clock
NCNT
N-2
N-1
N
N-1
N
N+1
Write
Write
NCR
N
W
N
W
Compare match
Waveform after noise cancellation
Capture status
Input capture is not performed
Input capture is performed

Figure 13.74 Conflict between Writing to NCR and Compare Match of NCR–NCNT (Example in Minimum Time-At-Level Cancellation Mode)

13.36.6 Conflict Regarding Down Counter D

Conflicts in DCNTDnm are described here.

(1) Conflict between Writing to DCNTDnm Counter and Count-Down

When a write to DCNTDnm occurs simultaneously with decrementation of the down counter, writing to DCNTDnm is performed. The attempt to decrement the value is ignored and decrementation recommences from the new value on the next count-down clock.

(2) Conflict between Writing to DCNTDnm Counter and Underflow

When a write to DCNTDnm occurs simultaneously with underflow, writing to DCNTDnm is performed. The example in the left half of figure 13.75 shows operation when a count-down clock is input simultaneously with a write to DCNTDnm when the DCNTDnm value is H'0000 00. Though the new value is written to DCNTDnm, count-down operation will be halted because underflow is detected. The underflow status flag is set. In the example shown by the waveforms in the right half of figure 13.75, DCNTDnm is written to one Pϕ cycle earlier so underflow is not detected.

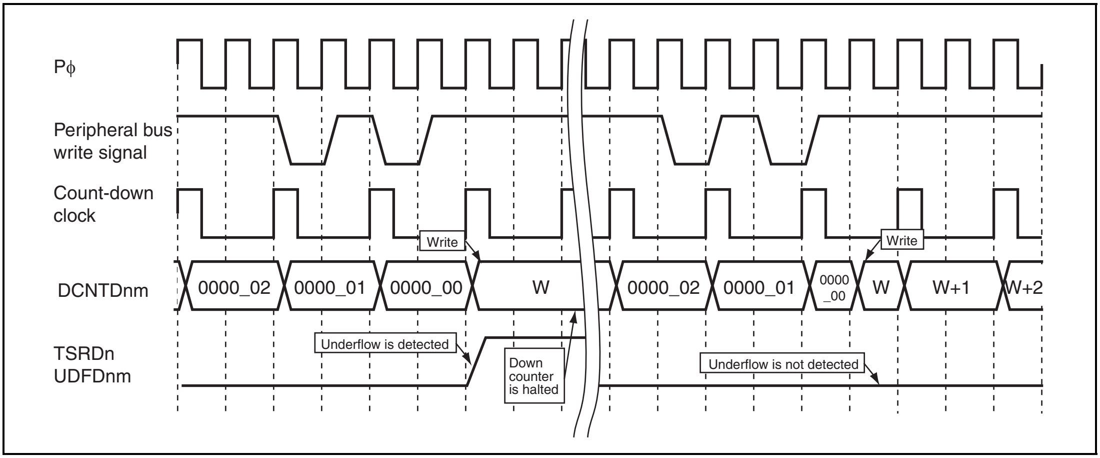


Figure 13.75 Conflict between Writing to DCNTDnm and Underflow

(3) Conflict between Writing to DCNTDnm Counter and Compare Match B (Counter Stop Trigger)

When a write to DCNTDnm occurs simultaneously with compare match B, writing to DCNTDnm is performed (if compare match B is selected as a condition to stop the down counter).

An example in which writing 0 to DCNTDnm occurs simultaneously with clearing of the counter by detection of compare match B is shown in the middle of figure 13.76. The attempt to clear DCNTDnm by compare match B is ignored and writing takes priority. However, the output on TODnmB is turned off due to compare match B, and DCNTDnm halts with the written value retained.

The waveforms in the right half of figure 13.76 show a case in which the write cycle occurs one Pϕ cycle earlier. During the Pϕ cycle subsequent to writing to DCNTDnm, the counter is cleared by compare match B.

The waveforms in the left half of figure 13.76 show an example in which writing is performed immediately after the counter has been cleared by compare match B.

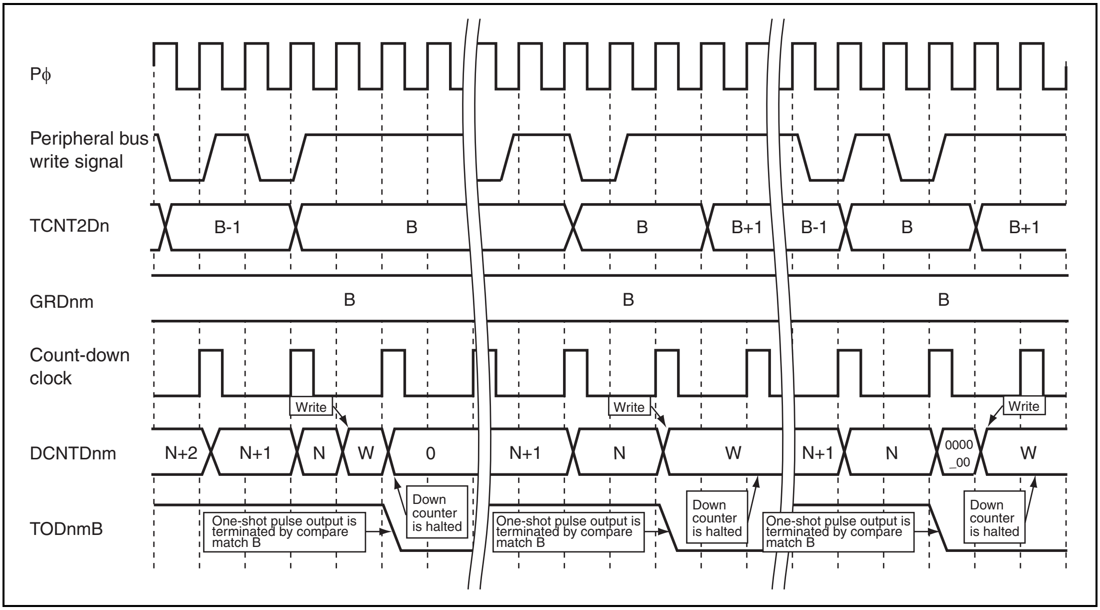

Pϕ
Peripheral bus write signal
TCNT2Dn
B-1
B
B
B+1
B-1
B
B+1
GRDnm
B
B
B
Count-down clock
Write
Write
Write
DCNTDnm
N+2
N+1
N
W
0
N+1
N
W
N+1
N
0000_00
W
TODnmB
One-shot pulse output is terminated by compare match B
Down counter is halted
One-shot pulse output is terminated by compare match B
Down counter is halted
One-shot pulse output is terminated by compare match B
Down counter is halted

Figure 13.76 Conflict between Writing to DCNTDnm and Counter Clearing by Compare Match B

(4) Conflict between Setting of Underflow Status Flag and Clearing by Writing 0

When clearing of the underflow status flag by writing 0 to it occurs simultaneously with underflow, status flag clearing takes priority. Shown below is an example in which setting of the status flag due to underflow of DCNTDnm occurs simultaneously with clearing of the status flag by 0 written to it (waveforms in the left half of figure 13.77). The waveforms in the right half of figure 13.77 show how the status flag is set again by underflow occurrence immediately after the status flag has been cleared.

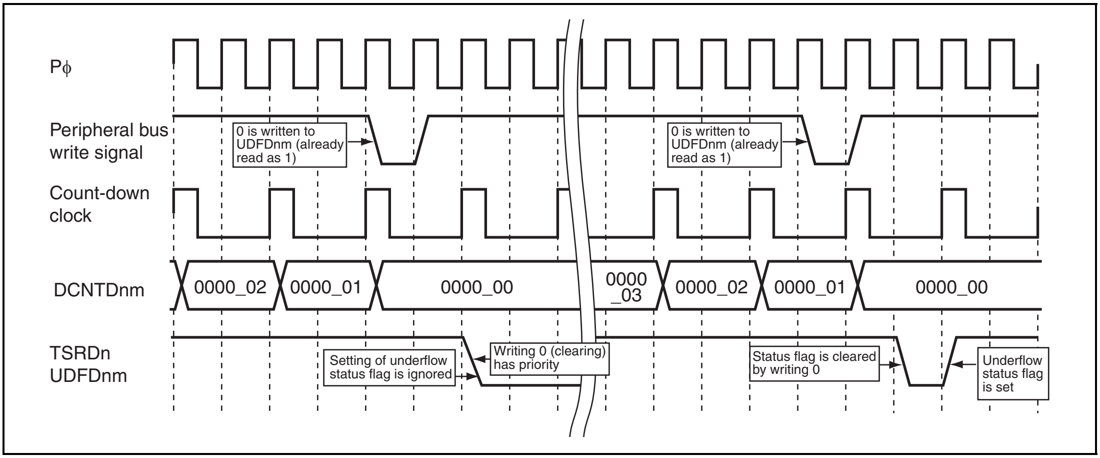

Pϕ
Peripheral bus write signal
0 is written to UDFDnm (already read as 1)
0 is written to UDFDnm (already read as 1)
Count-down clock
DCNTDnm
0000_02
0000_01
0000_00
0000 _03
0000_02
0000_01
0000_00
TSRDn UDFDnm
Setting of underflow status flag is ignored
Writing 0 (clearing) has priority
Status flag is cleared by writing 0
Underflow status flag is set

Figure 13.77 Conflict between Setting and Clearing of Underflow Status Flag

(5) Conflict between Setting of Underflow Status Flag and Clearing by DMAACK

When clearing of the underflow status flag by the DMAACK signal occurs simultaneously with underflow, status flag clearing takes priority.

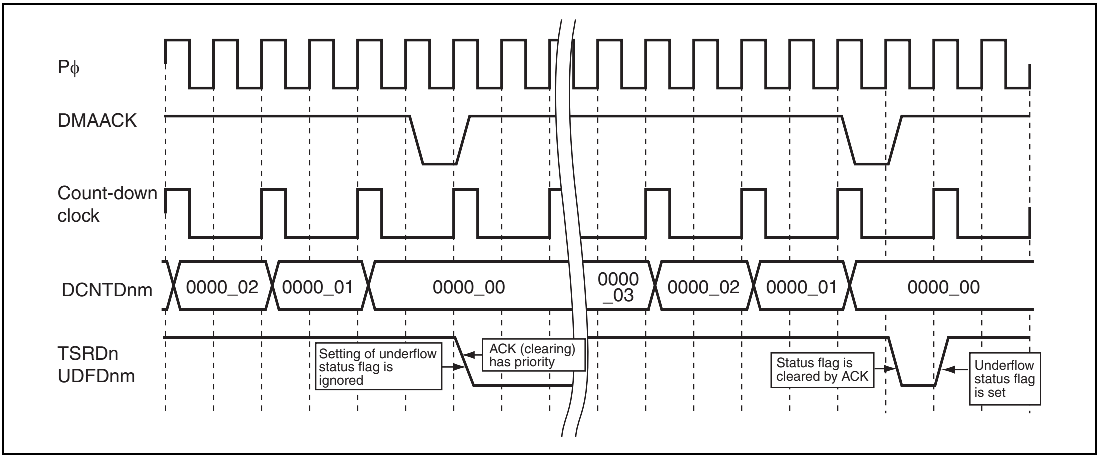

Pϕ
DMAACK
Count-down clock
DCNTDnm
0000_02
0000_01
0000_00
0000 _03
0000_02
0000_01
0000_00
TSRDn UDFDnm
Setting of underflow status flag is ignored
ACK (clearing) has priority
Status flag is cleared by ACK
Underflow status flag is set

Figure 13.78 Conflict between Setting of Underflow Status Flag and Clearing by DMAACK

(6) TODnmB Output at Occurrence of Down Counter Start Trigger When Down Counter Value is H'0000 00

The TODnmB output (one-shot pulse) is not started by down counter underflow.

(7) TODnmB Output at Simultaneous Occurrence of Start Trigger and Stop Trigger for Down Counter

When the down counter start trigger occurs simultaneously with the down counter stop trigger, the down counter is cleared to 0 by the stop trigger. In this case, the one-shot pulse output on the TODnmB pin is not started.

Similar to the above case, when the down counter start trigger occurs and then the down counter stop trigger occurs before the first count-down clock is input, the down counter is cleared to 0 without being decremented even once and the one-shot pulse output on the TODnmB pin is not started.

(8) Conflict between Down Counter Start Trigger and Underflow

When the down counter start trigger occurs simultaneously with underflow, DCNTDnm remains halted at the value of H'0000 00 (waveforms in the middle of figure 13.79). If the down counter had been in process of decrementing, the TODnmB output is turned off by underflow (waveforms in the left half of figure 13.79). If compare match A occurs while the down counter is halted (DCNTDnm = H'0000 00), the TODnmB output is kept negated (waveforms in the right half of figure 13.79). In any case, the underflow status flag is set at the same time the down counter start trigger is detected or in synchronization with the first count-down clock after trigger detection.

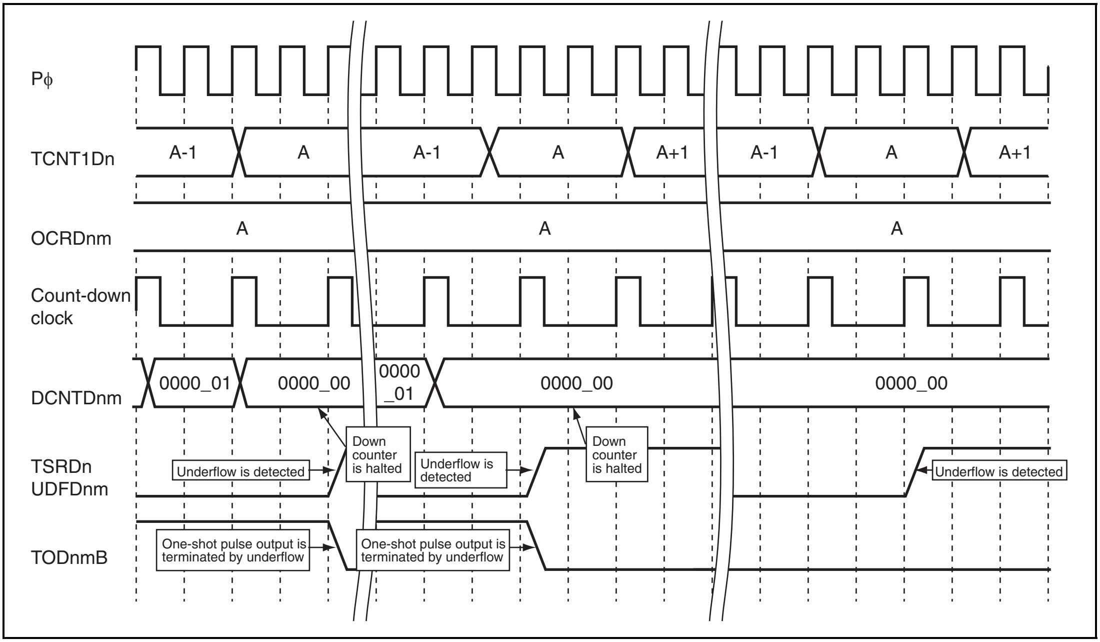


Figure 13.79 Conflict between Compare Match A and Underflow

13.36.7 Conflict between Timer B and Timer D

Conflicts between the counter clearing request from timer B and TCNT1Dn/TCNT2Dn are described here.

(1) Conflict between TCNT1Dn/TCNT2Dn Counter Clearing and Compare Match

Operation when clearing of the TCNT1Dn/TCNT2Dn counter by a counter clearing signal from timer B occurs simultaneously with compare match is shown below. The waveforms in the left half of figure 13.80 show a case in which the counter is cleared prior to compare match. On the other hand, the waveforms in the right half of figure 13.80 show a case in which the counter is cleared simultaneously with compare match.

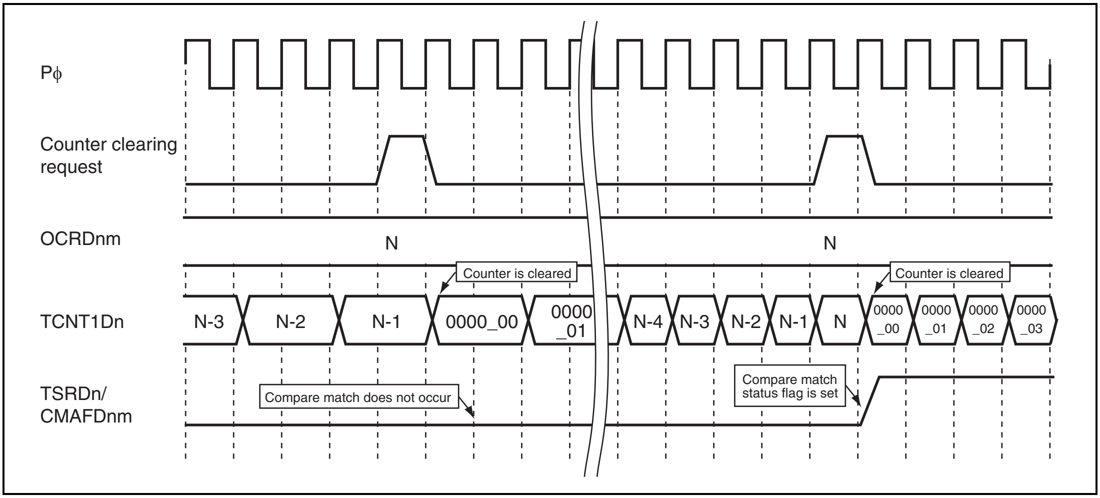


Figure 13.80 Conflict between Counter Clearing by Timer B and Compare Match

Table 13.43 Resources Related to Conflict Operation between Counter Clearing and Compare Match

Timer	Counter	Counter Clearing Source	Compare Match Register	Status
Timer D	TCNT1Dn	TCNT1Dn/TCNT2Dn clearing request from timer B	OCRDnm	CMFADnm
	TCNT2Dn		GRDnm	CMFBDnm

(2) Conflict between Writing to TCNT1Dn/TCNT2Dn Counter and Counter Clearing by Timer B

When a write to TCNT1Dn/TCNT2Dn occurs simultaneously with a counter clearing signal from timer B, the counter is not cleared but writing to the counter is performed (waveforms in the left half of figure 13.81). The waveforms in the right half of figure 13.81 show a case in which writing to TCNT1Dn is one Pϕ cycle later. This is the same for TCNT2Dn.

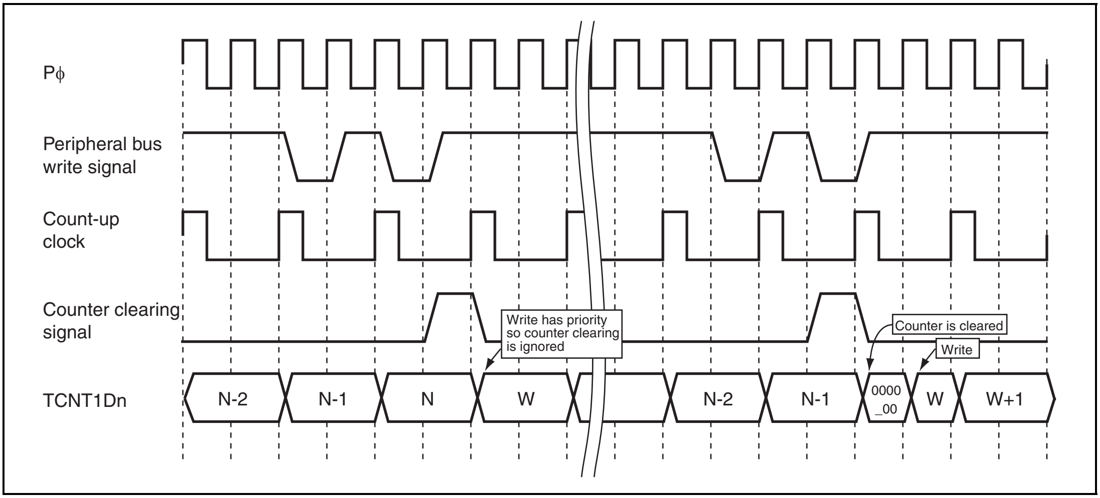


Figure 13.81 Conflict between Writing to TCNT1Dn and Counter Clearing

(3) Conflict between TCNT1Dn/TCNT2Dn Counter Overflow and Counter Clearing by Timer B

When TCNT1Dn overflow occurs simultaneously with clearing of TCNT1Dn from timer B, the counter value is cleared to H'0000 00 by the counter clearing signal. In this case, the overflow status flag is not set (only for when C1CEDn = 1). The same applies to TCNT2Dn overflow.

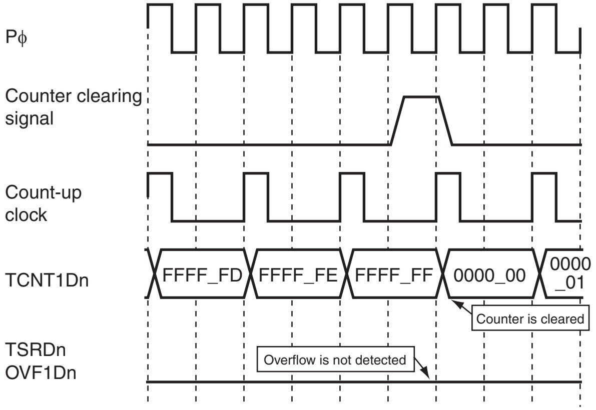


Figure 13.82 Conflict between Counter Clearing and Overflow

(4) Conflict between TCNT1Dn Clearing by Clearing Signal from Timer B and Input Capture to OSBRDn

Operation when clearing of the TCNT1Dn counter by timer B occurs simultaneously with input capture to an offset base register is shown below. When capture and counter clearing occur simultaneously, the counter value before clearing is captured in OSBRDn. At the same time, the TCNT1Dn counter is cleared (waveforms in the left half of figure 13.83).

The waveforms in the right half of figure 13.83 show a case in which capture is performed one Pϕ cycle after the counter has been cleared and the counter value after clearing is captured in OSBRDn.

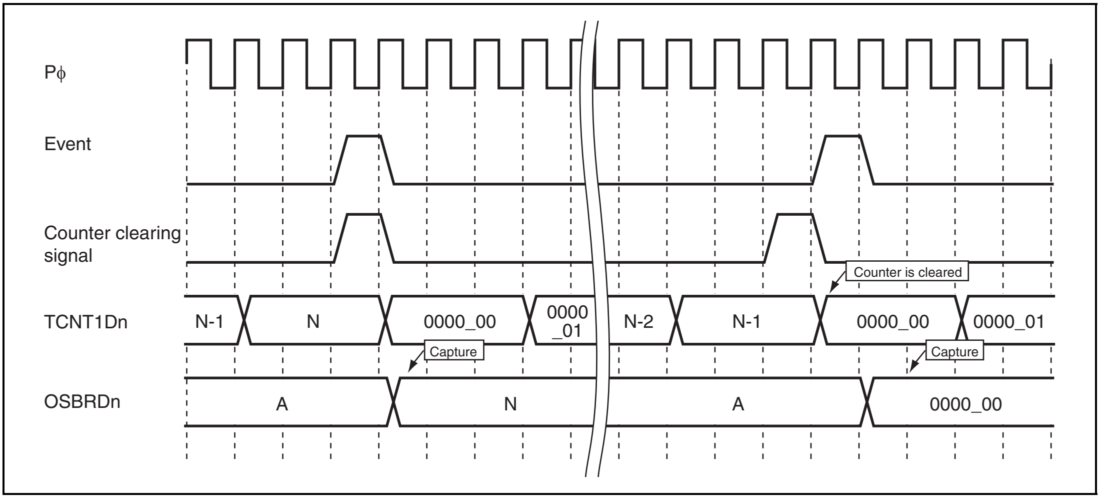


Figure 13.83 Conflict between TCNT1Dn Counter Clearing and Input Capture to OSBRDn

13.36.8 Compare-Match Operation Specification

The compare-match operation specification depends on each timer. Theoperation specifications can be grouped into three types, according to timing of compare-match occurrence and detection condition.

- Type 1

  A compare match occurs on the next Pϕ cycle after the value of a timer counter has reached the value set in the corresponding compare match register.
  Detection of compare-match is performed on each Pϕ cycle.
  Detection of compare-match is also performed on writing to the timer counter or compare-match register.
- Type 2

  A compare match occurs on the next Pϕ cycle after the value of a timer counter has reached the value set in the corresponding compare match register.
  Detection of compare-match is performed on each Pϕ cycle on which timer counters are incremented or decremented.
  Detection of compare-match is not performed on writing to the timer counter or compare-match register.
- Type 3

  A compare match occurs on the next counter clock cycle after the value of a timer counter has reached the value set in the corresponding compare match register.
  Detection of compare-match is performed on each counter clock cycle.
  Detection of compare-match is not performed on writing to the timer counter or compare-match register.

Table 13.44 Compare-match Operation Specification

Timer	Counter	Compare Match Register	Compare-match type
Timer B	TCNTB0	OCRB0	Type 3
	TCNTB1	OCRB1	Type 1
	TCNTB6	OCRB6/OCRB7	Type 3
Timer C	TCNTCn	GRCnm	Type 1
Timer D	TCNT1Dn	OCRDnm	Type 2
	TCNT2Dn	GRDnm	Type 2
Timer E	TCNTEnm	CYLREnm	Type 3
Timer F	ECNTAFn	GRAFn	Type 1
	ECNTBFn	GRBFn	Type 1
	ECNTCFn	GRBFn	Type 1
Timer G	TCNTGn	OCRGn	Type 1
Timer H	TCNT1H	OCR1H	Type 1
Timer J	TCNTJn	OCRJn	Type 1

Section 14 Watchdog Timer (WDT)

This LSI includes the watchdog timer (WDT) which externally outputs an overflow signal ($\overline{\text{WDTOVF}}$) on overflow of the counter when the value of the counter has not been updated because of a system malfunction. The WDT can simultaneously generate an internal reset signal for the entire LSI.

When this watchdog function is not needed, the WDT can be used as an interval timer. In the interval timer operation, an interval timer interrupt is generated at each counter overflow.

14.1 Features

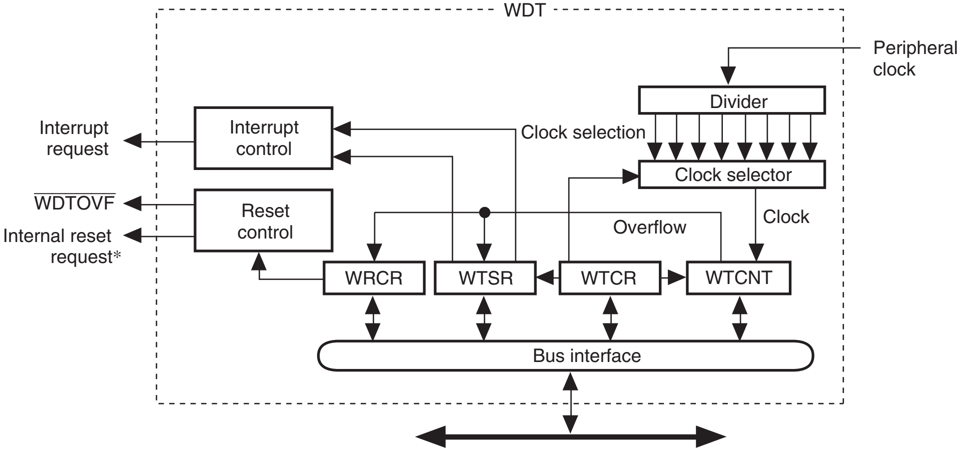
- Can switch between watchdog timer mode and interval timer mode.
- Outputs $\overline{\text{WDTOVF}}$ signal in watchdog timer mode

  When the counter overflows in watchdog timer mode, the $\overline{\text{WDTOVF}}$ signal is output externally. It is possible to select whether to reset the LSI internally when this happens.
- Interrupt generation in interval timer mode

  An interval timer interrupt is generated when the counter overflows.
- Choice of eight counter input clocks

  Eight clocks (Pϕ × 1 to Pϕ × 1/16384) that are obtained by dividing the peripheral clock can be selected.

Figure 14.1 is a block diagram of the WDT.



[Legend]

WTCR:	Watchdog timer control register
WTSR:	Watchdog timer status register
WRCR:	Watchdog reset control register
WTCNT:	Watchdog timer counter

Note: * The internal reset signal can be generated by register setting.

Figure 14.1 Block Diagram of WDT

14.2 Input/Output Pin

Table 14.1 shows the pin configuration of the WDT.

Table 14.1 Pin Configuration

Pin Name	Symbol	I/O	Function
Watchdog timer overflow	$\overline{\text{WDTOVF}}$	Output	Outputs the counter overflow signal in watchdog timer mode

14.3 Register Descriptions

The WDT has the following registers.

Table 14.2 Register Configuration

Register Name	Symbol	R/W	Initial Value	Address	Access Size
Watchdog timer control register	WTCR	R/W	H'0000	H'FFFE 0000	8, 16
Watchdog timer counter	WTCNT	R/W	H'0000	H'FFFE 0002	8, 16
Watchdog timer status register	WTSR	R/W	H'0000	H'FFFE 0004	8, 16
Watchdog reset control register	WRCR	R/W	H'0000	H'FFFE 0006	8, 16

14.3.1 Watchdog Timer Control Register (WTCR)

WTCR is a 16-bit readable/writable register that stores an 8-bit write key, selects the operating mode and the clock used for WTCNT counting, and enables the timer.

WTCR is initialized by a power-on reset caused by the $\overline{\text{RES}}$ pin, a transition to the hardware standby mode, or by a WTCNT overflow in watchdog timer mode.

WTCR must be written in words. When changing the WT/$\overline{\text{IT}}$, TME, and CKS bits, write H'A5 to the TCRKEY bits simultaneously. Writing a value other than H'A5 to the TCRKEY bits or writing in bytes are ignored.

WTCR can be read in words or bytes. Note that data written to the TCRKEY bits is not retained; therefore, the TCRKEY bits are always read as H'00.

Note: The method for writing to this register differs from that for other registers to prevent erroneous writes. See section 14.3.5, Notes on Register Access, for details.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TCRKEY[7:0]								-	WT/$\overline{\text{IT}}$	TME	-	-	CKS[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R	R/W	R/W	R	R	R/W	R/W	R/W

Note: * Data written to these bits is not retained.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	TCRKEY [7:0]	H'00	R/W*	WTCR Write Key Code These bits enable or disable changing the WT/$\overline{\text{IT}}$, TME, and CKS bits. Since data written to these bits is not retained, these bits are always read as H'00. H'A5: Enables changing the WT/$\overline{\text{IT}}$, TME, and CKS bits Other than H'A5: Disables changing the WT/$\overline{\text{IT}}$, TME, and CKS bits
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6	WT/$\overline{\text{IT}}$	0	R/W	Timer Mode Select Selects whether to use the WDT as a watchdog timer or an interval timer. 0: WDT used as interval timer 1: WDT used as watchdog timer Note: When WTCNT overflows in watchdog timer mode, the $\overline{\text{WDTOVF}}$ signal is output externally. If this bit is modified when the WDT is running, the up-count may not be performed correctly.

<table>
<tr><th>Bit</th><th>Bit Name</th><th>Initial Value</th><th>R/W</th><th>Description</th></tr>
<tr><td>5</td><td>TME</td><td>0</td><td>R/W</td><td>WTCNT Enable
Starts or stops WTCNT counting.
0: Counting disabled
Counting up stops and WTCNT value is stored
1: Counting enabled</td></tr>
<tr><td>4, 3</td><td>—</td><td>All 0</td><td>R</td><td>Reserved
These bits are always read as 0. The write value should always be 0.</td></tr>
<tr><td>2 to 0</td><td>CKS[2:0]</td><td>000</td><td>R/W</td><td>Clock Select
These bits select the clock to be used for the WTCNT counting from the eight types obtainable by dividing the peripheral clock (Pϕ). The overflow cycle shown below is the value when the peripheral clock (Pϕ) is 20 MHz. The table below shows the overflow cycles on condition that Pϕ is 20 or 40 MHz (the time taken until the counter overflows when Pϕ is 40 MHz is the same as in cases where Pϕ is 20 MHz because the clock signal divided by 2 will be used).
<table>
<tr><th>Clock Division Ratio</th><th>Overflow Cycle</th></tr>
<tr><th></th><th>Pϕ = 20 or 40 MHz</th></tr>
<tr><td>000: 1 × Pϕ</td><td>12.8 µs</td></tr>
<tr><td>001: 1/64 × Pϕ</td><td>819.2 µs</td></tr>
<tr><td>010: 1/128 × Pϕ</td><td>1.6 ms</td></tr>
<tr><td>011: 1/256 × Pϕ</td><td>3.3 ms</td></tr>
<tr><td>100: 1/512 × Pϕ</td><td>6.6 ms</td></tr>
<tr><td>101: 1/1024 × Pϕ</td><td>13.1 ms</td></tr>
<tr><td>110: 1/4096 × Pϕ</td><td>52.4 ms</td></tr>
<tr><td>111: 1/16384 × Pϕ</td><td>209.7 ms</td></tr>
</table>
Note: If bits CKS[2:0] are changed during the WTCNT counting, counting up may not be performed correctly. Ensure that the WTCNT is stopped to change these bits.</td></tr>
</table>

Note: * Data written to these bits is not retained.

14.3.2 Watchdog Timer Counter (WTCNT)

WTCNT is a 16-bit readable/writable register that stores an 8-bit write key and an 8-bit counter value. Setting the TME bit in WTCR to 1 starts counting on the internal clock selected by the CKS[2:0] bits in WTCR.

When a WTCNT overflow occurs, it generates a watchdog timer overflow signal ($\overline{\text{WDTOVF}}$) in watchdog timer mode and an interval timer interrupt (ITI) in interval timer mode. WTCNT is initialized to H'0000 by a power-on reset caused by the $\overline{\text{RES}}$ pin, a transition to the hardware standby mode, or a WTCNT overflow in watchdog timer mode.

WTCNT must be written in words. When changing the TCNT bits, write H'5A to the TCNTKEY bits simultaneously. Writing a value other than H'5A to the TCNTKEY bits or writing in bytes are ignored.

WTCNT can be read in words or bytes. Note that data written to the TCNTKEY bits is not retained, therefore, the TCNTKEY bits are always read as H'00.

Note: The method for writing to this register differs from that for other registers to prevent erroneous writes. See section 14.3.5, Notes on Register Access, for details.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TCNTKEY[7:0]								TCNT[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Data written to these bits is not retained.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	TCNTKEY [7:0]	H'00	R/W*	WTCNT Write Key Code These bits enable or disable changing the TCNT bits. Since data written to these bits is not retained, these bits are always read as H'00. H'5A: Enables changing the TCNT bits Other than H'A5: Disables changing the TCNT bits
7 to 0	TCNT[7:0]	H'00	R/W	8-Bit Timer Counter Value An overflow is generated when these bits change from H'FF to H'00.

Note: * Data written to these bits is not retained.

14.3.3 Watchdog Timer Status Register (WTSR)

WTSR is a 16-bit readable/writable register that stores an 8-bit write key, an overflow flag in watchdog timer mode, and an overflow flag in interval timer mode.

WTSR is initialized to H'0000 by a power-on reset caused by the $\overline{\text{RES}}$ pin or a transition to the hardware standby mode. It is not initialized by a WTCNT overflow in watchdog timer mode or by an internal reset caused by a WTCNT overflow.

WTSR must be written in words. When changing the WOVF and IOVF bits, write H'A5 to the TSRKEY bits simultaneously. Writing a value other than H'A5 to the TSRKEY bits or writing in bytes are ignored.

WSTR can be read in words or bytes. Note that data written to the TSRKEY bits is not retained, therefore, the TSRKEY bits are always read as H'00.

Note: The method for writing to this register differs from that for other registers to prevent erroneous writes. See section 14.3.5, Notes on Register Access, for details.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSRKEY[7:0]								WOVF	-	-	-	IOVF	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*2	R	R	R	R/W*2	R	R	R

Notes: 1. Data written to these bits is not retained.
2. Only 0 can be written to clear the flag. Writing 1 to the flag is invalid.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	TSRKEY [7:0]	H'00	R/W*1	WTCR Write Key Code These bits enable or disable changing the WOVF and IOVF bits. Since data written to these bits is not retained, these bits are always read as H'00. H'A5: Enables changing the WOVF and IOVF bits. Other than H'A5: Disables changing the WOVF and IOVF bits.

Bit	Bit Name	Initial Value	R/W	Description
7	WOVF	0	R/W*[2]	Watchdog Timer Overflow Indicates that WTCNT has overflowed in watchdog timer mode. This bit is not set to 1 in interval timer mode. 0: WTCNT has not overflowed 1: WTCNT has overflowed in watchdog timer mode [Clearing condition] • When 0 is written to this bit after reading it as 1
6 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	IOVF	0	R/W*[2]	Interval Timer Overflow Indicates that WTCNT has overflowed in interval timer mode. This bit is not set to 1 in watchdog timer mode. 0: WTCNT has not overflowed 1: WTCNT has overflowed in interval timer mode [Clearing condition] • When 0 is written to this bit after reading it as 1
2 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Notes: 1. Data written to these bits is not retained.

2. Only 0 can be written to clear the flag. Writing 1 to the flag is invalid.

14.3.4 Watchdog Reset Control Register (WRCR)

WRCR is a 16-bit readable/writable register that stores an 8-bit write key and enables or disables an internal reset caused by a WTCNT overflow.

WRCR is initialized to H'0000 by a power-on reset caused by the $\overline{\text{RES}}$ pin or a transition to the hardware standby mode. It is not initialized by a WTCNT overflow in watchdog timer mode or by an internal reset caused by a WTCNT overflow.

WRCR must be written in words. When changing the RSTE bit, write H'5A to the RCRKEY bits simultaneously. Writing a value other than H'5A to the RCRKEY bits or writing in bytes are ignored.

WRCR can be read in words or bytes. Note that data written to the RCRKEY bits is not retained, therefore, the RCRKEY bits are always read as H'00.

Note: The method for writing to WRCR differs from that for other registers to prevent erroneous writes. See section 14.3.5, Notes on Register Access, for details.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RCRKEY[7:0]								RSTE	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W	R	R	R	R	R	R	R

Note: * Data written to these bits is not retained.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	RCRKEY [7:0]	H'00	R/W*	WRCR Write Key Code These bits enable or disable changing the RSTE bit. Since data written to these bits is not retained, these bits are always read as H'00. H'5A: Enables changing the RSTE bit. Other than H'A5: Disables changing the RSTE bit.

Bit	Bit Name	Initial Value	R/W	Description
7	RSTE	0	R/W	Reset Enable Selects whether to generate a signal to initialize the LSI internally if WTCNT overflows in watchdog timer mode. In interval timer mode, this setting is ignored. 0: This LSI is not initialized when WTCNT overflows* 1: This LSI is initialized when WTCNT overflows Note: This LSI is not initialized internally, but WTCNT and WTCR are initialized by an overflow of WTCNT.
6 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: * Data written to these bits is not retained.

14.3.5 Notes on Register Access

The watchdog timer counter (WTCNT), watchdog timer control register (WTCR), watchdog timer status register (WTSR), and watchdog reset control register (WRCR) are more difficult to write to than other registers. The procedures for reading or writing to these registers are given below.

(1) Writing to WTCNT, WTCR, WTSR, and WRCR

These registers must be written by a word transfer instruction. They cannot be written by a byte transfer instruction.

When writing to WTCNT and WRCR, set the upper byte to H'5A and the lower byte to the write data as shown in figure 14.2 and then transfer data. When writing to WTCR and WTSR, set the upper byte to H'A5 and the lower byte to the write data and then transfer data. This transfer procedure writes the lower byte data to WTCNT, WTCR, WTSR, or WRCR.

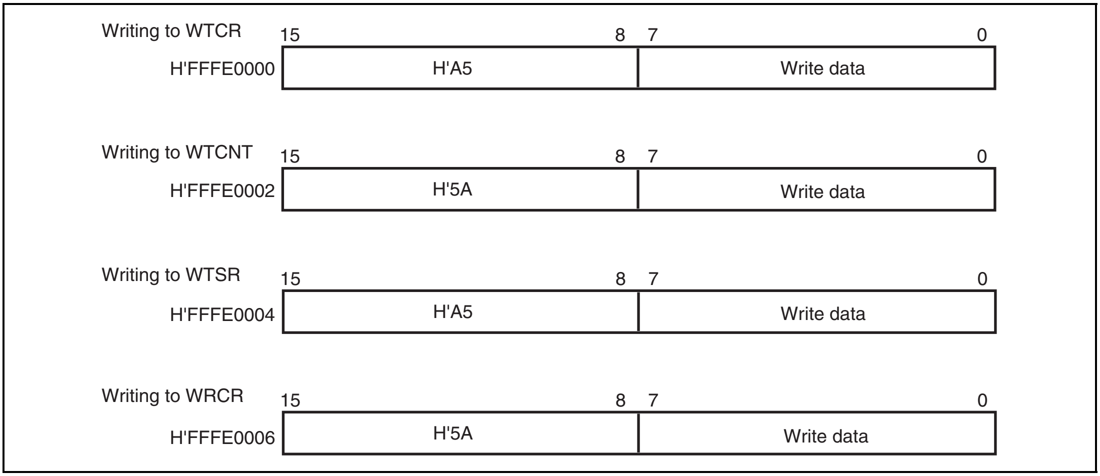


Figure 14.2 Writing to WTCNT, WTCR, WTSR, and WRCR

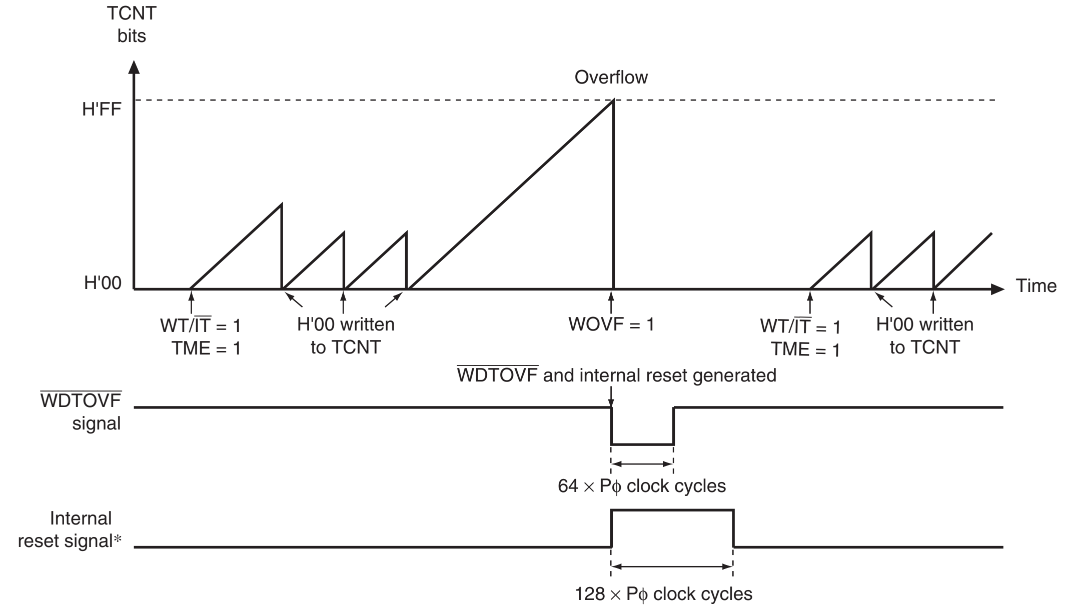
(2) Reading from WTCNT, WTCR, WTSR, and WRCR

These registers are read in a method similar to other registers. Both byte and word transfer instructions can be used.

14.4 WDT Usage

14.4.1 Using WDT in Watchdog Timer Mode

1. Set the WT/$\overline{\mathrm{IT}}$ bit in WTCR to 1 (watchdog timer mode). Set the CKS[2:0] bits in WTCR (the type of counting clock), the RSTE bit in WRCR (whether to initialize this LSI on a WTCNT overflow), and WTCNT to the initial value of the counter.
2. Set the TME bit in WTCR to 1 to start counting in watchdog timer mode.
3. While operating in watchdog timer mode, rewrite the counter periodically to H'00 to prevent the counter from overflowing.
4. When the counter overflows, the WDT sets the WOVF bit in WTCR to 1, and the $\overline{\mathrm{WDTOVF}}$ signal is output externally (figure 14.3). The $\overline{\mathrm{WDTOVF}}$ signal can be used to initialize the system. The $\overline{\mathrm{WDTOVF}}$ signal is output for 64 × Pϕ clock cycles.
5. If the RSTE bit in WRCR is set to 1, a signal to initialize the inside of this LSI can be generated simultaneously with the $\overline{\mathrm{WDTOVF}}$ signal. The internal reset signal is output for 128 × Pϕ clock cycles.
6. When a WTCNT overflow reset is generated simultaneously with a reset input on the $\overline{\mathrm{RES}}$ pin, the $\overline{\mathrm{RES}}$ pin reset takes priority, and the WOVF bit in WTSR is cleared to 0.



[Legend]

WT/$\overline{\text{IT}}$: Timer mode select bit
TME: Timer enable bit

Note: * Internal reset signal is generated only when the RSTE bit is set to 1.

Figure 14.3 Operation in Watchdog Timer Mode

14.4.2 Using WDT in Interval Timer Mode

When using the WDT in interval timer mode, an interval timer interrupt (ITI) is generated every overflow of the counter. This enables interrupts to be generated at specified periods.

1. Clear the WT/$\overline{\text{IT}}$ bit in WTCR to 0. Set the CKS[2:0] bits in WTCR (the type of counting clock) and the TCNT bits in WTCNT to the initial value of the counter.
2. Set the TME bit in WTCR to 1 to start the counting in interval timer mode.
3. When the counter overflows, the WDT sets the IOVF bit in WTCR to 1 and an interval timer interrupt (ITI) is requested to the INTC. The counter then resumes counting.

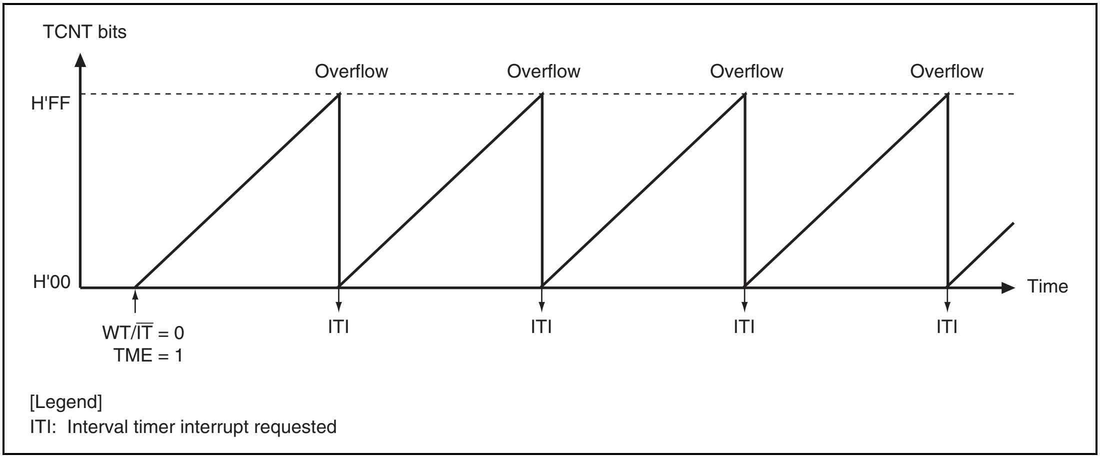


Figure 14.4 Operation in Interval Timer Mode

14.5 Usage Notes

Pay attention to the following points when using the WDT in either the interval or watchdog timer mode.

14.5.1 Timer Error

It takes one cycle of the counting clock until WTCNT counts up for the first time after the TME bit in WTCR is set to start, meaning that it takes one cycle of the Pϕ clock at a minimum and one cycle of the divided Pϕ clock at worst (the division ratio is selected by the CKS[2:0] bits). The timer error is the duration from when leaving the power-on reset state to the first counting-up timing after the TME bit is set. The timing of subsequent counting up depends on the selected frequency division ratio.

This also applies to the timing of the first counting up after WTCNT has been written to during timer operation.

14.5.2 Changing of Division Ratio

If the CKS[2:0] bits in WTCR are changed during WTCNT counting, correct operation may not be guaranteed. Change these bits after stopping the WDT (after clearing the TME bit to 0).

14.5.3 Switching between Watchdog and Interval Timer Modes

If the watchdog and interval timer modes are switched during WDT operation, correct operation may not be guaranteed. Switch timer modes after stopping the WDT (after clearing the TME bit to 0).

14.5.4 System Reset by $\overline{\text{WDTOVF}}$ Signal

If the $\overline{\text{WDTOVF}}$ signal is input on the $\overline{\text{RES}}$ pin of this LSI, this LSI cannot be initialized correctly.

Avoid input of the $\overline{\text{WDTOVF}}$ signal to the $\overline{\text{RES}}$ pin of this LSI through glue logic circuits. To initialize the entire system with the $\overline{\text{WDTOVF}}$ signal, use the circuit shown in figure 14.5.

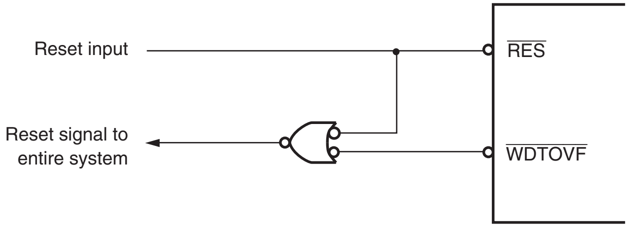


Figure 14.5 Example of System Reset Circuit Using $\overline{\text{WDTOVF}}$ Signal

Section 15 Compare Match Timer (CMT)

This LSI has an on-chip compare match timer (CMT) consisting of a two-channel 16-bit timer. The CMT has a16-bit counter, and can generate interrupts at set intervals.

15.1 Features

- Independent selection of four counter input clocks at two channels

  Any of four internal clocks (Pϕ/8, Pϕ/32, Pϕ/128, and Pϕ/512) can be selected.
- Selection of DMA transfer request or interrupt request generation on compare match by DMAC setting

Figure 15.1 shows a block diagram of CMT.

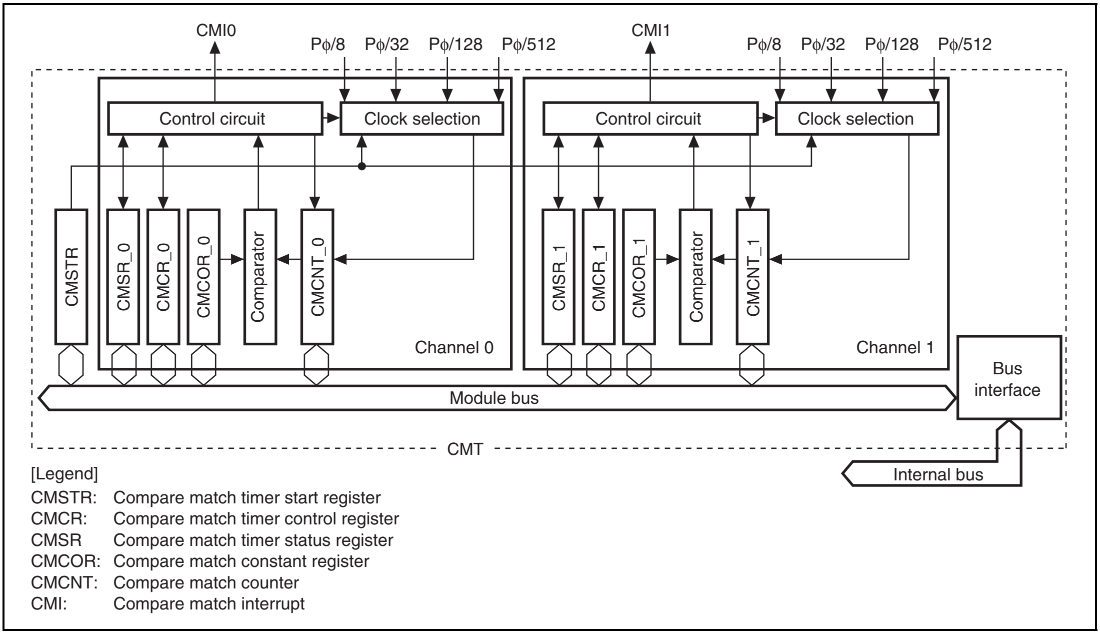


Figure 15.1 Block Diagram of CMT

15.2 Register Descriptions

The CMT has the following registers.

Table 15.1 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Common	Compare match timer start register	CMSTR	R/W	H'0000	H'FFFEC000	16
0	Compare match timer control register_0	CMCR_0	R/W	H'00	H'FFFEC010	8
	Compare match timer status register_0	CMSR_0	R/(W)*	H'00	H'FFFEC011	8
	Compare match counter_0	CMCNT_0	R/W	H'0000	H'FFFEC012	16
	Compare match constant register_0	CMCOR_0	R/W	H'FFFF	H'FFFEC014	16
1	Compare match timer control register_1	CMCR_1	R/W	H'00	H'FFFEC020	8
	Compare match timer status register_1	CMSR_1	R/(W)*	H'00	H'FFFEC021	8
	Compare match counter_1	CMCNT_1	R/W	H'0000	H'FFFEC022	16
	Compare match constant register_1	CMCOR_1	R/W	H'FFFF	H'FFFEC024	16

15.2.1 Compare Match Timer Start Register (CMSTR)

CMSTR is a 16-bit register that selects whether or not the compare match counter (CMCNT) operates.

CMSTR is initialized to H'0000 by a power-on reset or in standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	STR1	STR0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	STR1	0	R/W	Count Start 1 Specifies whether or not compare match counter 1 operates. 0: Stops counting by CMCNT_1* 1: Starts counting by CMCNT_1
0	STR0	0	R/W	Count Start 0 Specifies whether or not compare match counter 0 operates. 0: Stops counting by CMCNT_0* 1: Starts counting by CMCNT_0

Note: * The value in CMCNT is retained when counting stops.

15.2.2 Compare Match Timer Control Register (CMCR)

CMCR is an 8-bit register that enables interrupts and DMA transfer requests, and selects the counter input clock.

CMCR is initialized to H'00 by a power-on reset or entry into standby mode.

Bit:	7	6	5	4	3	2	1	0
	-	CMIE	-	-	-	-	CKS[1:0]	
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6	CMIE	0	R/W	Compare Match Interrupt Enable Enables or disables compare match interrupt (CMI) generation when CMCNT and CMCOR values match (CMF = 1). 0: Disables the compare match interrupt (CMI) 1: Enables the compare match interrupt (CMI)
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	CKS[1:0]	00	R/W	Clock Select These bits select the clock to be input to CMCNT among four types of internal clocks, which are obtained by dividing the peripheral clock (Pϕ). When the STR bit in CMSTR is set to 1, CMCNT starts counting on the clock selected with bits CKS1 and CKS0. 00: Pϕ/8 01: Pϕ/32 10: Pϕ/128 11: Pϕ/512

15.2.3 Compare Match Timer Status Register (CMSR)

CMSR is an 8-bit register that indicates compare match generation.

CMSR is initialized to H'00 by a power-on reset or in standby mode.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	CMF
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/(W)*

Note: * Only 0 can be written to clear the flag after 1 is read.

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	CMF	0	R/(W)*[1]	Compare Match Flag Indicates whether or not the values of CMCNT and CMCOR match. 0: CMCNT and CMCOR values do not match [Clearing condition] • When 0 is written to CMF after reading CMF = 1 1: CMCNT and CMCOR values match*[2]

Notes: 1. Only 0 can be written to clear the flag after 1 is read.

2. The CMF bit is also set to 1 when the values in CMCNT and CMCOR match by writing the same value to CMCNT as in CMCOR.

15.2.4 Compare Match Counter (CMCNT)

CMCNT is a 16-bit register used as an up-counter. When the counter input clock is selected with bits CKS1 and CKS0 in CMCR, and the STR bit in CMSTR is set to 1, CMCNT starts counting using the selected clock. When the value in CMCNT and the value in compare match constant register (CMCOR) match, CMCNT is cleared to H'0000 and the CMF flag in CMSR is set to 1.

CMCNT is initialized to H'0000 by a power-on reset or in standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15.2.5 Compare Match Constant Register (CMCOR)

CMCOR is a 16-bit register that sets the interval up to a compare match with CMCNT.

CMCOR is initialized to H'FFFF by a power-on reset or in standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15.3 Operation

15.3.1 Interval Count Operation

When an internal clock is selected with the CKS1 and CKS0 bits in CMCR and the STR bit in CMSTR is set to 1, CMCNT starts incrementing using the selected clock. When the values in CMCNT and CMCOR match, CMCNT is cleared to H'0000 and the CMF flag in CMSR is set to 1. When the CMIE bit in CMCR is set to 1 at this time, a compare match interrupt (CMI) is requested. CMCNT then starts counting up again from H'0000.

Figure 15.2 shows the operation of the compare match counter.

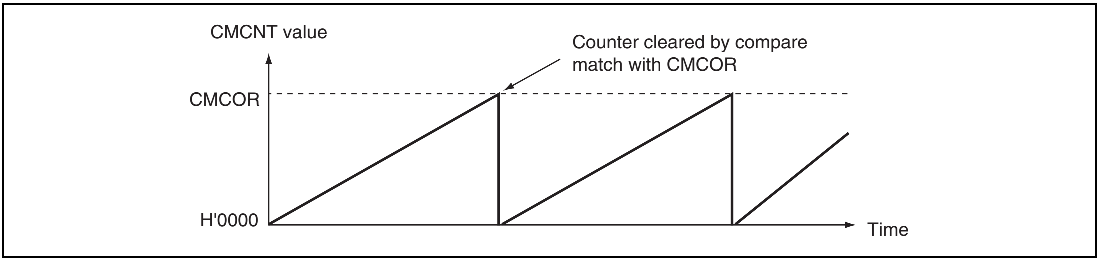


Figure 15.2 Counter Operation

15.3.2 CMCNT Count Timing

One of four clocks (Pϕ/8, Pϕ/32, Pϕ/128, and Pϕ/512) obtained by dividing the peripheral clock (Pϕ) can be selected with the CKS[1:0] bits in CMCR. Figure 15.3 shows the timing.

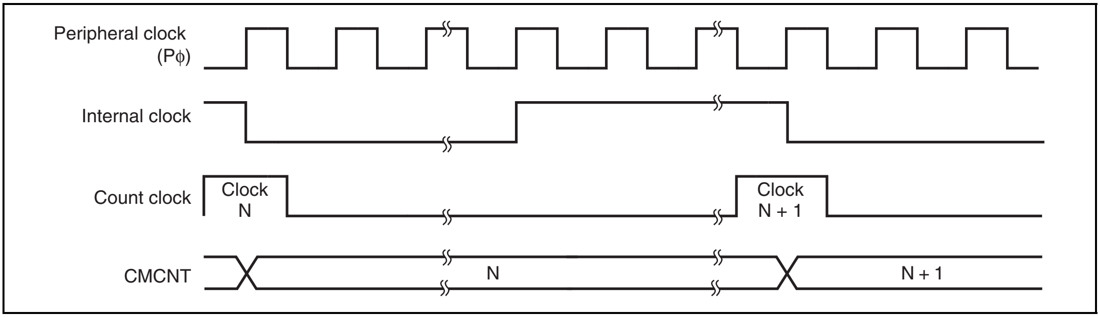


Figure 15.3 Count Timing

15.4 Interrupts

15.4.1 Interrupt Sources and DMA Transfer Requests

The CMT has channels and each of them to which a different vector address is allocated has a compare match interrupt. When both the interrupt request flag (CMF) and the interrupt enable bit (CMIE) are set to 1, the corresponding interrupt request is output. When the interrupt is used to activate a CPU interrupt, the priority of channels can be changed by the interrupt controller settings. For details, see section 8, Interrupt Controller (INTC).

Clear the CMF bit to 0 by the user exception handling routine. If this operation is not carried out, another interrupt will be generated. The direct memory access controller (DMAC) can be set to be activated when a compare match interrupt is requested. In this case, an interrupt is not issued to the CPU. If the setting to activate the DMAC has not been made, an interrupt request is sent to the CPU. The CMF bit is automatically cleared to 0 when data is transferred by the DMAC.

15.4.2 Timing of Compare Match Flag Setting

When CMCOR and CMCNT match, a compare match signal is generated and the CMF bit in CMSR is set to 1. The compare match signal is generated in the last state in which the values match (when the CMCNT value is updated to H'0000). That is, after a match between CMCOR and CMCNT, the compare match signal is not generated until the next CMCNT counter clock input. Figure 15.4 shows the timing of CMF bit setting.

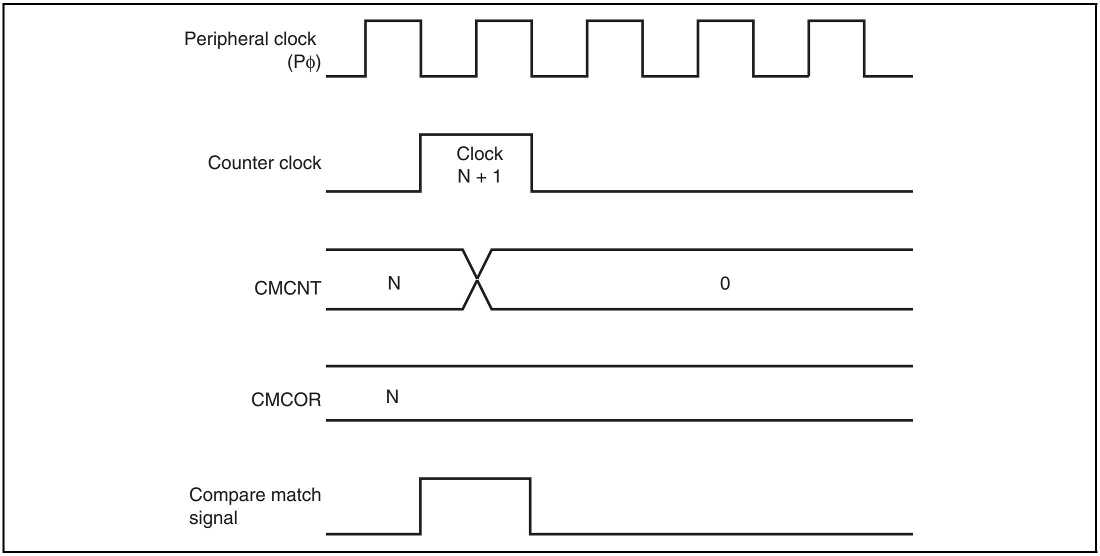


Figure 15.4 Timing of CMF Setting

15.4.3 Timing of Compare Match Flag Clearing

The CMF bit in CMSR is cleared by first, reading as 1 then writing to 0. However, in the case of the DMAC being activated, the CMF bit is automatically cleared to 0 when data is transferred by the DMAC.

Note that the CMF bit is cleared regardless of the state of counter operation. Even when counting is not in progress, the CMF bit is cleared by writing to the CMSR register by the CPU or upon generation of an ACK signal from the DMAC.

15.5 Usage Notes

15.5.1 Conflict between Write and Compare-Match Processes of CMCNT

When the compare match signal is generated in the T2 cycle while writing to CMCNT, clearing CMCNT has priority over writing to it. In this case, CMCNT is not written to. Figure 15.5 shows the timing to clear the CMCNT counter.

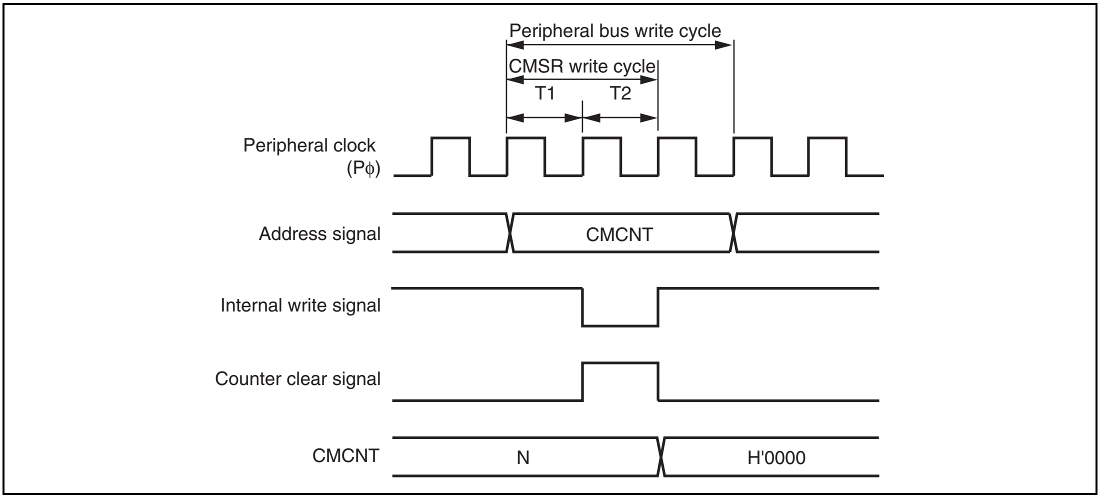


Figure 15.5 Conflict between Write and Compare Match Processes of CMCNT

15.5.2 Conflict between Word-Write and Count-Up Processes of CMCNT

Even when the count-up occurs in the T2 cycle while writing to CMCNT in words, the writing has priority over the count-up. In this case, the count-up is not performed. Figure 15.6 shows the timing to write to CMCNT in words.

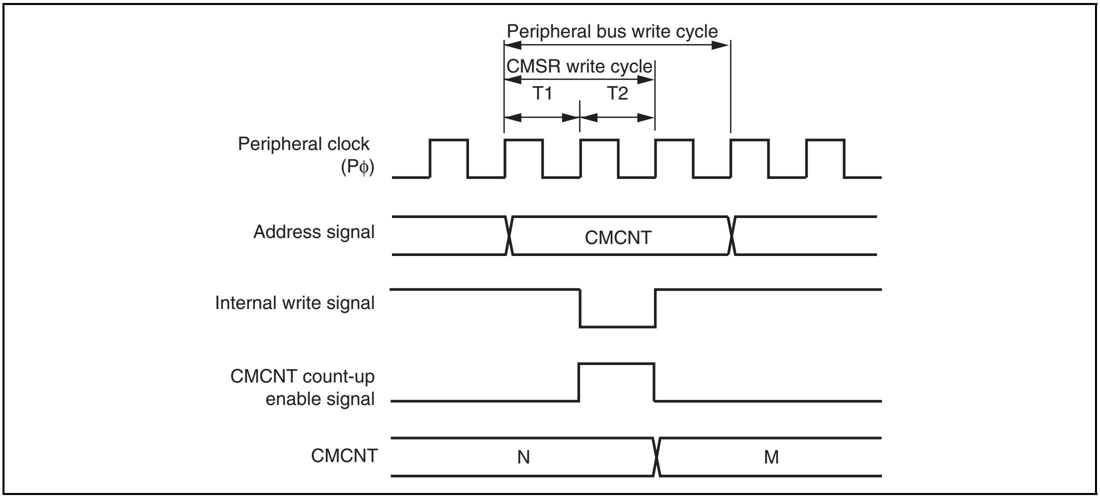


Figure 15.6 Conflict between Word-Write and Count-Up Processes of CMCNT

15.5.3 Conflict between Setting of Compare-Match Flag and Clearing by the CPU

If, while the compare-match flag is set, clearing of the flag by the CPU (during T2 cycle within CMSR register write cycle), that is, writing a 0 to the flag after reading a 1 from it, and setting of the flag on a compare match coincide, setting of the flag is given priority. If a compare-match occurs during the period between reading a 1 from and writing a 0 to the flag, clearing by writing a 0 will not be done.

Figure 15.7 shows the timing explained above. The left shows the case where setting and clearing the flag coincide, and the right shows the case where a compare-match occurs after reading a 1 from the flag and before writing a 0 to it.

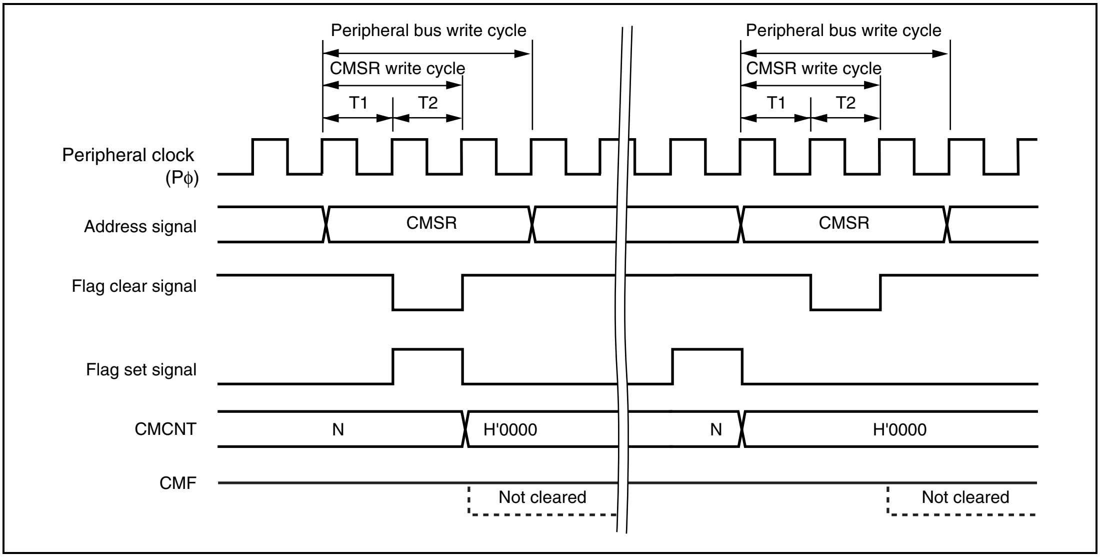


Figure 15.7 Conflict between Compare-Match Flag Setting and Clearing by the CPU

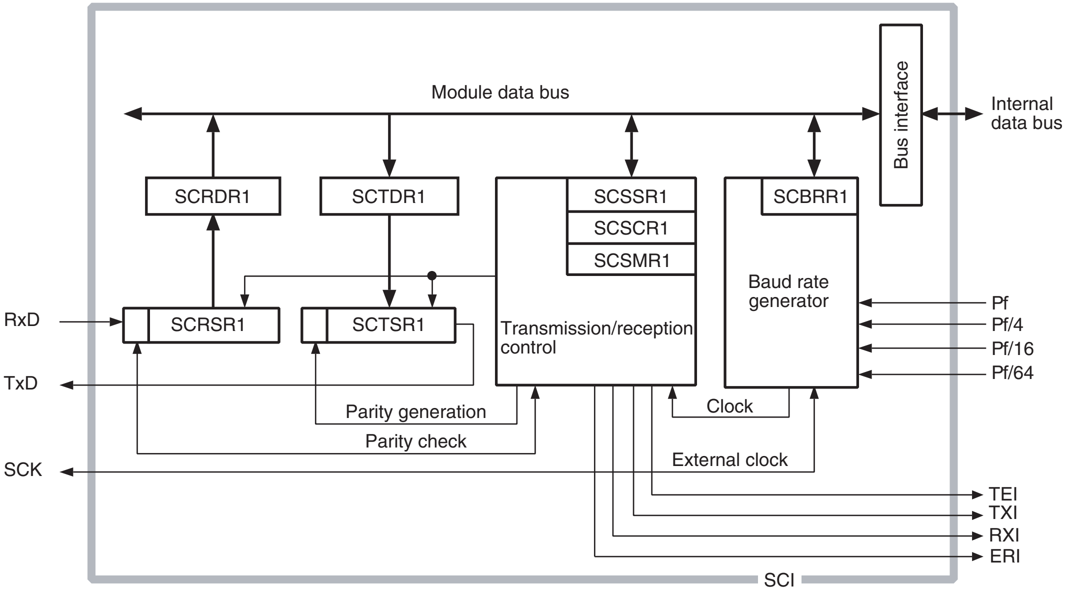
Section 16 Serial Communications Interface (SCI)

The SCI can handle both asynchronous and clock synchronous serial communications.

16.1 Features

- Choice of asynchronous or clock synchronous serial communications mode
- Asynchronous mode:
  - Serial data communications are performed by start-stop in character units. The SCI can communicate with a universal asynchronous receiver/transmitter (UART), an asynchronous communications interface adapter (ACIA), or any other communications chip that employs a standard asynchronous serial system. There are twelve selectable serial data communication formats.
  - Data length: 7 or 8 bits
  - Stop bit length: 1 or 2 bits
  - Parity: Even, odd, or none
  - Receive error detection: Parity, overrun, and framing errors
  - Break detection: Break is detected by reading the RxD pin level directly when a framing error occurs.
- Clock synchronous mode:
  - Serial data communications are synchronized with a clock signal. The SCI can communicate with other chips having a clock synchronous communications function.
  - Data length: 8 bits
  - Receive error detection: Overrun errors
- Full duplex communications: The transmitting and receiving sections are independent, so the SCI can transmit and receive simultaneously. Both sections use double buffering, so high-speed continuous data transfer is possible in both the transmit and receive directions.
- On-chip baud rate generator with selectable bit rates
- Internal or external transmit/receive clock source in clock synchronous mode: From either baud rate generator (internal clock) or SCK pin (external clock)
- Four types of interrupts: There are four interrupt sources, transmit-data-empty, transmit end, receive-data-full, and receive error interrupts, and each interrupt can be requested independently. The automotive direct memory access controller (A-DMAC) can be activated by the transmit-data-empty interrupt or receive-data-full interrupt to transfer data.

Figure 16.1 shows a block diagram of the SCI.



[Legend]
SCRSR1: Receive shift register
SCRDR1: Receive data register
SCTSR1: Transmit shift register
SCTDR1: Transmit data register
SCSMR1: Serial mode register
SCSCR1: Serial control register
SCSSR1: Serial status register
SCBRR1: Bit rate register

Figure 16.1 Block Diagram of SCI

16.2 Input/Output Pins

Table 16.1 shows the pin configuration of the SCI.

Table 16.1 Pin Configuration

Channel	Pin Name*	Symbol	I/O	Function
A	Serial clock pins	SCK_A	I/O	Clock input/output
	Receive data pins	RxD_A	Input	Receive data input
	Transmit data pins	TxD_A	Output	Transmit data output
B	Serial clock pins	SCK_B	I/O	Clock input/output
	Receive data pins	RxD_B	Input	Receive data input
	Transmit data pins	TxD_B	Output	Transmit data output
C	Serial clock pins	SCK_C	I/O	Clock input/output
	Receive data pins	RxD_C	Input	Receive data input
	Transmit data pins	TxD_C	Output	Transmit data output
D	Serial clock pins	SCK_D	I/O	Clock input/output
	Receive data pins	RxD_D	Input	Receive data input
	Transmit data pins	TxD_D	Output	Transmit data output
E	Serial clock pins	SCK_E	I/O	Clock input/output
	Receive data pins	RxD_E	Input	Receive data input
	Transmit data pins	TxD_E	Output	Transmit data output

Notes: 1. These pins serve as serial pins if the SCI operation modes are set appropriately by the TE, RE, and CKE1 bits in SCSCR1 and the $C/\overline{A}$ bit in SCSMR1.

2. Pin names SCK, RxD, and TxD are used in the description for all channels, omitting the channel designation.

16.3 Register Description

The SCI has the following registers. These registers specify the asynchronous or clock synchronous mode, data format, bit rate, and control the transmitter or the receiver. For details on register addresses and register states during each processing state, refer to section 33, List of Registers.

Table 16.2 Register Configuration

Channel	Register Name	Abbreviation*[1]	R/W	Initial Value	Address	Access Size
A	Serial mode register	SCSMR1A	R/W	H'00	H'FFFF8000	8
	Bit rate register	SCBRR1A	R/W	H'FF	H'FFFF8004	8
	Serial control register	SCSCR1A	R/W	H'00	H'FFFF8008	8
	Transmit data register	SCTDR1A	R/W	H'FF	H'FFFF800C	8
	Serial status register	SCSSR1A	R/(W)*[2]	H'84	H'FFFF8010	8
	Receive data register	SCRDR1A	R	H'00	H'FFFF8014	8
B	Serial mode register	SCSMR1B	R/W	H'00	H'FFFF8800	8
	Bit rate register	SCBRR1B	R/W	H'FF	H'FFFF8804	8
	Serial control register	SCSCR1B	R/W	H'00	H'FFFF8808	8
	Transmit data register	SCTDR1B	R/W	H'FF	H'FFFF880C	8
	Serial status register	SCSSR1B	R/(W)*[2]	H'84	H'FFFF8810	8
	Receive data register	SCRDR1B	R	H'00	H'FFFF8814	8
C	Serial mode register	SCSMR1C	R/W	H'00	H'FFFF9000	8
	Bit rate register	SCBRR1C	R/W	H'FF	H'FFFF9004	8
	Serial control register	SCSCR1C	R/W	H'00	H'FFFF9008	8
	Transmit data register	SCTDR1C	R/W	H'FF	H'FFFF900C	8
	Serial status register	SCSSR1C	R/(W)*[2]	H'84	H'FFFF9010	8
	Receive data register	SCRDR1C	R	H'00	H'FFFF9014	8

Channel	Register Name	Abbreviation*[1]	R/W	Initial Value	Address	Access Size
D	Serial mode register	SCSMR1D	R/W	H'00	H'FFFF9800	8
	Bit rate register	SCBRR1D	R/W	H'FF	H'FFFF9804	8
	Serial control register	SCSCR1D	R/W	H'00	H'FFFF9808	8
	Transmit data register	SCTDR1D	R/W	H'FF	H'FFFF980C	8
	Serial status register	SCSSR1D	R/(W)*[2]	H'84	H'FFFF9810	8
	Receive data register	SCRDR1D	R	H'00	H'FFFF9814	8
E	Serial mode register	SCSMR1E	R/W	H'00	H'FFFFA000	8
	Bit rate register	SCBRR1E	R/W	H'FF	H'FFFFA004	8
	Serial control register	SCSCR1E	R/W	H'00	H'FFFFA008	8
	Transmit data register	SCTDR1E	R/W	H'FF	H'FFFFA00C	8
	Serial status register	SCSSR1E	R/(W)*[2]	H'84	H'FFFFA010	8
	Receive data register	SCRDR1E	R	H'00	H'FFFFA014	8

Notes: 1. Register names and abbreviations are used in the description for all channels, omitting the channel designation.

2. Writing only 0 is enabled to clear the flag.

16.3.1 Receive Shift Register (SCRSR1)

SCRSR1 receives serial data. Data input at the RxD pin is loaded into SCRSR1 in the order received, LSB (bit 0) first, converting the data to parallel form. When one byte has been received, it is automatically transferred to SCRDR1. The CPU cannot read or write to SCRSR1 directly.

Bit:	7	6	5	4	3	2	1	0
Initial value:	-	-	-	-	-	-	-	-
R/W:	-	-	-	-	-	-	-	-

16.3.2 Receive Data Register (SCRDR1)

SCRDR1 is a register that stores serial receive data. After receiving one byte of serial data, the SCI transfers the received data from the receive shift register (SCRSR1) into SCRDR1 for storage and completes operation. After that, SCRSR1 is ready to receive data.

Since SCRSR1 and SCRDR1 work as a double buffer in this way, data can be received continuously.

SCRDR1 is a read-only register and cannot be written to by the CPU.

SCRDR1 is initialized to H'00 by a power on reset or in standby mode.

Bit:	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

16.3.3 Transmit Shift Register (SCTSR1)

SCTSR1 transmits serial data. The SCI loads transmit data from the transmit data register (SCTDR1) into SCTSR1, then transmits the data serially from the TxD pin, LSB (bit 0) first. After transmitting one data byte, the SCI automatically loads the next transmit data from SCTDR1 into SCTSR1 and starts transmitting again. If the TDRE flag in the serial status register (SCSSR1) is set to 1, the SCI does not transfer data from SCTDR1 to SCTSR1. The CPU cannot read or write to SCTSR1 directly.

Bit:	7	6	5	4	3	2	1	0
Initial value:	-	-	-	-	-	-	-	-
R/W:	-	-	-	-	-	-	-	-

16.3.4 Transmit Data Register (SCTDR1)

SCTDR1 is an 8-bit register that stores data for serial transmission. When the SCI detects that the transmit shift register (SCTSR1) is empty, it moves transmit data written in the SCTDR1 into SCTSR1 and starts serial transmission. If the next transmit data has been written to SCTDR1 during serial transmission from SCTSR1, the SCI can transmit data continuously. SCTDR1 can always be written or read to by the CPU.

SCTDR1 is initialized to H'FF by a power on reset or in standby mode.

Bit:	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

16.3.5 Serial Mode Register (SCSMR1)

SCSMR1 is an 8-bit register that specifies the SCI serial communication format and selects the clock source for the baud rate generator.

The CPU can always read and write to SCSMR1.

SCSMR1 is initialized to H'00 by a power on reset or in standby mode.

Bit:	7	6	5	4	3	2	1	0
	$C/\overline{A}$	CHR	PE	$O/\overline{E}$	STOP	-	CKS[1:0]	
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	-	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	$C/\overline{A}$	0	R/W	Communication Mode Selects whether the SCI operates in asynchronous or clock synchronous mode. 0: Asynchronous mode 1: Clock synchronous mode

Bit	Bit Name	Initial Value	R/W	Description
6	CHR	0	R/W	Character Length Selects 7-bit or 8-bit data in asynchronous mode. In the clock synchronous mode, the data length is always eight bits, regardless of the CHR bit setting. When 7-bit data is selected, the MSB (bit 7) of the transmit data register (SCTDR1) is not transmitted. 0: 8-bit data 1: 7-bit data
5	PE	0	R/W	Parity Enable Selects whether to add a parity bit to transmit data and to check the parity of receive data, in asynchronous mode. In clock synchronous mode, a parity bit is neither added nor checked, regardless of the PE setting. 0: Parity bit not added or checked 1: Parity bit added and checked* Note: * When PE is set to 1, an even or odd parity bit is added to transmit data, depending on the parity mode (O/$\overline{\text{E}}$) setting. Receive data parity is checked according to the even/odd (O/$\overline{\text{E}}$) mode setting.
4	O/$\overline{\text{E}}$	0	R/W	Parity Mode Selects even or odd parity when parity bits are added and checked. The O/$\overline{\text{E}}$ setting is used only in asynchronous mode and only when the parity enable bit (PE) is set to 1 to enable parity addition and checking. The O/$\overline{\text{E}}$ setting is ignored in clock synchronous mode, or in asynchronous mode when parity addition and checking is disabled. 0: Even parity 1: Odd parity If even parity is selected, the parity bit is added to transmit data to make an even number of 1s in the transmitted character and parity bit combined. Receive data is checked to see if it has an even number of 1s in the received character and parity bit combined. If odd parity is selected, the parity bit is added to transmit data to make an odd number of 1s in the transmitted character and parity bit combined. Receive data is checked to see if it has an odd number of 1s in the received character and parity bit combined.

Bit	Bit Name	Initial Value	R/W	Description
3	STOP	0	R/W	Stop Bit Length Selects one or two bits as the stop bit length in asynchronous mode. This setting is used only in asynchronous mode. It is ignored in clock synchronous mode because no stop bits are added. 0: One stop bit*[1] 1: Two stop bits*[2] When receiving, only the first stop bit is checked, regardless of the STOP bit setting. If the second stop bit is 1, it is treated as a stop bit, but if the second stop bit is 0, it is treated as the start bit of the next incoming character. Notes: 1. When transmitting, a single 1-bit is added at the end of each transmitted character. 2. When transmitting, two 1 bits are added at the end of each transmitted character.
2	—	0	—	Reserved This bit is always read as 0. The write value should always be 0.
1, 0	CKS[1:0]	00	R/W	Clock Select 1 and 0 Select the internal clock source of the on-chip baud rate generator. Four clock sources are available. Pϕ, Pϕ/4, Pϕ/16 and Pϕ/64. For further information on the clock source, bit rate register settings, and baud rate, see section 16.3.8, Bit Rate Register (SCBRR1). 00: Pϕ 01: Pϕ/4 10: Pϕ/16 11: Pϕ/64 Note: Pϕ: Peripheral clock

16.3.6 Serial Control Register (SCSCR1)

SCSCR1 is an 8-bit register that enables or disables SCI transmission/reception and interrupt requests and selects the transmit/receive clock source. The CPU can always read and write to SCSCR1.

SCSCR1 is initialized to H'00 by a power on reset or in standby mode.

Bit:	7	6	5	4	3	2	1	0
	TIE	RIE	TE	RE	-	TEIE	CKE1	-
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	-	R/W	R/W	-

Bit	Bit Name	Initial Value	R/W	Description
7	TIE	0	R/W	Transmit Interrupt Enable Enables or disables a transmit-data-empty interrupt (TXI) to be issued when the TDRE flag in the serial status register (SCSSR1) is set to 1 after serial transmit data is sent from the transmit data register (SCTDR1) to the transmit shift register (SCTSR1). 0: Transmit-data-empty interrupt request (TXI) is disabled* 1: Transmit-data-empty interrupt request (TXI) is enabled Note: * TXI can be canceled by clearing the TDRE flag to 0 after reading TDRE = 1 or by clearing the TIE bit to 0. TXI can also be canceled when data is written to SCTDR1 through the A-DMAC.

Bit	Bit Name	Initial Value	R/W	Description
6	RIE	0	R/W	Receive Interrupt Enable Enables or disables a receive-data-full interrupt (RXI) and a receive error interrupt (ERI) to be issued when the RDRF flag in SCSSR1 is set to 1 after the serial data received is transferred from the receive shift register (SCRSR1) to the receive data register (SCRDR1). 0: Receive-data-full interrupt (RXI) and receive-error interrupt (ERI) requests are disabled* 1: Receive-data-full interrupt (RXI) and receive-error interrupt (ERI) requests are enabled Note: * RXI can be canceled by clearing the RDRF flag after reading RDRF = 1 or by clearing the RIE bit to 0. RXI can also be canceled when data is read from SCRDR1 through the A-DMAC. ERI can be canceled by clearing the FER, PER, and ORER flags after reading FER, PER, or ORER = 1 or by clearing the RIE bit to 0.
5	TE	0	R/W	Transmit Enable Enables or disables the SCI serial transmitter. To disable the transmitter and a transmit-data-full interrupt (TXI) simultaneously, clear the TE and TIE bits to 0 simultaneously. 0: Transmitter disabled*1 1: Transmitter enabled*2 Notes: 1. The TDRE flag in SCSSR1 is fixed at 1. 2. Serial transmission starts after writing transmit data into SCTDR1 and clearing the TDRE flag in SCSSR1 to 0 while the transmitter is enabled. Select the transmit format in the serial mode register (SCSMR1) before setting TE to 1.

Bit	Bit Name	Initial Value	R/W	Description
4	RE	0	R/W	Receive Enable Enables or disables the SCI serial receiver. 0: Receiver disabled*1 1: Receiver enabled*2 Notes: 1. Clearing RE to 0 does not affect the receive flags (RDRF, FER, PER, and ORER). These flags retain their previous values. To disable the receiver and a receive-data-full interrupt (RXI) simultaneously, clear the RE and RIE bits to 0 simultaneously. 2. Serial reception starts when a start bit is detected in asynchronous mode, or synchronous clock input is detected in clock synchronous mode. Select the receive format in SCSMR1 before setting RE to 1.
3	—	0	—	Reserved This bit is always read as 0. The write value should always be 0.
2	TEIE	0	R/W	Transmit End Interrupt Enable Enables or disables a transmit end interrupt (TEI) to be issued when no valid transmit data is found in SCTDR1 during MSB data transmission. TEI can be canceled by clearing the TEND flag to 0 (by clearing the TDRE flag in SCSSR1 to 0 after reading TDRE = 1) or by clearing the TEIE bit to 0. TEI can also be canceled when data is written to SCTDR1 through the A-DMAC. 0: Transmit end interrupt request (TEI) is disabled 1: Transmit end interrupt request (TEI) is enabled

Bit	Bit Name	Initial Value	R/W	Description
1	CKE1	0	R/W	Clock Enable 1 Selects the SCI clock source and specifies the SCK pin functions in SCI clock synchronous mode. The $C/\overline{A}$ bit in SCSMR1 should be set to 1 before setting the CKE1 bit. For details on clock source selection, refer to table 16.11 in section 16.4, Operation. 0: Internal clock, SCK pin used for synchronous clock output 1: External clock, SCK pin used for clock input* Note: * Do not set the CKE1 bit to 1 in asynchronous mode ($C/\overline{A}$ = 0). Otherwise, correct operation cannot be guaranteed.
0	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

16.3.7 Serial Status Register (SCSSR1)

SCSSR1 is an 8-bit register that contains status flags to indicate the SCI operating state.

The CPU can always read and write to SCSSR1, but cannot write 1 to status flags TDRE, RDRF, ORER, PER, and FER. These flags can be cleared to 0 only after 1 is read from the flags. The TEND flag is a read-only bit and cannot be modified.

SCSSR1 is initialized to H'84 by a power on reset or in standby mode.

Bit:	7	6	5	4	3	2	1	0
	TDRE	RDRF	ORER	FER	PER	TEND	-	-
Initial value:	1	0	0	0	0	1	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	-	-

Note: * Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

Bit	Bit Name	Initial Value	R/W	Description
7	TDRE	1	R/(W)*	Transmit Data Register Empty Indicates whether data has been transferred from the transmit data register (SCTDR1) to the transmit shift register (SCTSR1) and SCTDR1 has become ready to be written with next serial transmit data. 0: Indicates that SCTDR1 holds valid transmit data [Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When data is written to SCTDR1 through the A-DMAC 1: Indicates that SCTDR1 does not hold valid transmit data [Setting conditions] • By a power-on reset • When the TE bit in SCSCR1 is 0 • When data is transferred from SCTDR1 to SCTSR1 and data can be written to SCTDR1

Bit	Bit Name	Initial Value	R/W	Description
6	RDRF	0	R/(W)*	Receive Data Register Full Indicates that the received data is stored in the receive data register (SCRDR1). 0: Indicates that valid received data is not stored in SCRDR1 [Clearing conditions] • By a power-on reset • When 0 is written to RDRF after reading RDRF = 1 • When data is read from SCRDR1 through the A-DMAC 1: Indicates that valid received data is stored in SCRDR1 [Setting condition] • When serial reception ends normally and receive data is transferred from SCRSR1 to SCRDR1 Note: SCRDR1 and the RDRF flag are not affected and retain their previous states even if an error is detected during data reception or if the RE bit in the serial control register (SCSCR1) is cleared to 0. If reception of the next data is completed while the RDRF flag is still set to 1, an overrun error will occur and the received data will be lost.

Bit	Bit Name	Initial Value	R/W	Description
5	ORER	0	R/(W)*	Overrun Error Indicates that an overrun error occurred during reception, causing abnormal termination. 0: Indicates that reception is in progress or was completed successfully*1 [Clearing conditions] • By a power-on reset • When 0 is written to ORER after reading ORER = 1 1: Indicates that an overrun error occurred during reception*2 [Setting condition] • When the next serial reception is completed while RDRF = 1 Notes: 1. The ORER flag is not affected and retains its previous value when the RE bit in SCSCR1 is cleared to 0. 2. The receive data prior to the overrun error is retained in SCRDR1, and the data received subsequently is lost. Subsequent serial reception cannot be continued while the ORER flag is set to 1.

Bit	Bit Name	Initial Value	R/W	Description
4	FER	0	R/(W)*	Framing Error Indicates that a framing error occurred during data reception in asynchronous mode, causing abnormal termination. 0: Indicates that reception is in progress or was completed successfully*1 [Clearing conditions] • By a power-on reset • When 0 is written to FER after reading FER = 1 1: Indicates that a framing error occurred during reception [Setting condition] • When the SCI founds that the stop bit at the end of the received data is 0 after completing reception*2 Notes: 1. The FER flag is not affected and retains its previous value when the RE bit in SCSCR1 is cleared to 0. 2. In 2-stop-bit mode, only the first stop bit is checked for a value to 1; the second stop bit is not checked. If a framing error occurs, the receive data is transferred to SCRDR1 but the RDRF flag is not set. Subsequent serial reception cannot be continued while the FER flag is set to 1.

Bit	Bit Name	Initial Value	R/W	Description
3	PER	0	R/(W)*	Parity Error Indicates that a parity error occurred during data reception in asynchronous mode, causing abnormal termination. 0: Indicates that reception is in progress or was completed successfully*[1] [Clearing conditions] • By a power-on reset • When 0 is written to PER after reading PER = 1 1: Indicates that a parity error occurred during reception*[2] [Setting condition] • When the number of 1s in the received data and parity does not match the even or odd parity specified by the O/$\overline{\text{E}}$ bit in the serial mode register (SCSMR1). Notes: 1. The PER flag is not affected and retains its previous value when the RE bit in SCSCR1 is cleared to 0. 2. If a parity error occurs, the receive data is transferred to SCRDR1 but the RDRF flag is not set. Subsequent serial reception cannot be continued while the PER flag is set to 1.

Bit	Bit Name	Initial Value	R/W	Description
2	TEND	1	R	Transmit End Indicates that no valid data was in SCTDR1 during transmission of the last bit of the transmit character and transmission has ended. The TEND flag is read-only and cannot be modified. 0: Indicates that transmission is in progress [Clearing condition] • When 0 is written to TDRE after reading TDRE = 1 1: Indicates that transmission has ended [Setting conditions] • By a power-on reset • When the TE bit in SCSCR1 is 0 • When TDRE = 1 during transmission of the last bit of a 1-byte serial transmit character Note: Do not use the TEND flag as a transmit end flag when the A-DMAC writes data to SCTDR1 in response to a TXI interrupt request.
1, 0	—	All 0	—	Reserved These bits are always read as 0. The write value should always be 0.

Note: * Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

16.3.8 Bit Rate Register (SCBRR1)

SCBRR1 is an 8-bit register that, together with the baud rate generator clock source selected by the CKS1 and CKS0 bits in the serial mode register (SCSMR1), determines the serial transmit/receive bit rate.

The CPU can always read and write to SCBRR1.

SCBRR1 is initialized to H'FF by a power on reset or in standby mode. The maximum bit rate in the asynchronous mode is 2.5 Mbps. Do not set more than the maximum in SCBRR1. Otherwise, normal operation cannot be guaranteed.

The SCBRR1 setting is calculated as follows:

Bit:	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- Asynchronous mode:

$$N = \frac{P\phi}{32 \times 2^{2n-1} \times B} \times 10^6 - 1$$

- Clock synchronous mode:

$$N = \frac{P\phi}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$$

B: Bit rate (bits/s)

N: SCBRR1 setting for baud rate generator ($0 \leq N \leq 255$)
(The setting value should satisfy the electrical characteristics.)

Pϕ: Operating frequency for peripheral modules (MHz)

n: Baud rate generator clock source (n = 0, 1, 2, 3) (for the clock sources and values of n, see table 16.3.)

Table 16.3 SCSMR1 Settings

		SCSMR1 Settings	
n	Clock Source	CKS1	CKS0
0	Pϕ	0	0
1	Pϕ/4	0	1
2	Pϕ/16	1	0
3	Pϕ/64	1	1

Note: The bit rate error in asynchronous is given by the following formula:

$$\text{Error (\%)} = \left\{ \frac{P\phi \times 10^6}{(N + 1) \times B \times 32 \times 2^{2n-1}} - 1 \right\} \times 100$$

Table 16.4 shows examples of SCBRR1 settings in asynchronous mode, and table 16.5 shows examples of SCBRR1 settings in clock synchronous mode.

Table 16.4 Bit Rates and SCBRR1 Settings in Asynchronous Mode

	Pϕ (MHz)					
	16			18		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)
110	3	141	0.03	3	159	-0.12
150	3	103	0.16	3	116	0.16
300	3	51	0.16	2	233	0.16
600	3	25	0.16	2	116	0.16
1200	3	12	0.16	1	233	0.16
2400	2	25	0.16	1	116	0.16
4800	2	12	0.16	0	233	0.16
9600	1	25	0.16	0	116	0.16
19200	1	12	0.16	0	58	-0.69
31250	1	7	0	0	35	0
38400	0	25	0.16	0	28	1.02

	Pϕ (MHz)					
	20			32		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)
110	3	177	-0.25	—	—	—
150	3	129	0.16	3	207	0.16
300	3	64	0.16	3	103	0.16
600	2	129	0.16	2	207	0.16
1200	2	64	0.16	2	103	0.16
2400	1	129	0.16	1	207	0.16
4800	1	64	0.16	1	103	0.16
9600	0	129	0.16	0	207	0.16
19200	0	64	0.16	0	103	0.16
31250	0	39	0	0	63	0
38400	0	32	-1.36	0	51	0.16

	Pϕ (MHz)					
	36			40		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)
110	—	—	—	—	—	—
150	3	233	0.16	—	—	—
300	3	116	0.16	3	129	0.16
600	2	233	0.16	3	64	0.16
1200	2	116	0.16	2	129	0.16
2400	1	233	0.16	2	64	0.16
4800	1	116	0.16	1	129	0.16
9600	0	233	0.16	1	64	0.16
19200	0	116	0.16	0	129	0.16
31250	0	71	0	0	79	0
38400	0	58	-0.68	0	64	0.16

Table 16.5 Bit Rates and SCBRR1 Settings in Clock Synchronous Mode

	Pϕ (MHz)							
	16		18		20		32	
Bit Rate (bits/s)	**n**	**N**	**n**	**N**	**n**	**N**	**n**	**N**
10	—	—	—	—	—	—	—	—
250	3	249	—	—	—	—	—	—
500	3	124	—	—	—	—	3	249
1k	2	249	—	—	—	—	3	124
2.5k	2	99	—	—	2	124	2	199
5k	1	199	1	224	1	249	2	99
10k	1	99	—	—	1	124	1	199
25k	0	159	1	44	1	49	1	79
50k	0	79	0	89	1	24	0	159
100k	0	39	0	44	0	49	0	79
250k	0	15	0	17	0	19	0	31
500k	0	7	0	8	0	9	0	15
1M	0	3	—	—	0	4	0	7
2.5M	—	—	—	—	0	1	—	—
5M	—	—	—	—	—	—	—	—

	Pϕ (MHz)			
	36		**40**	
Bit Rate (bits/s)	**n**	**N**	**n**	**N**
10	—	—	—	—
250	—	—	—	—
500	—	—	—	—
1k	—	—	—	—
2.5k	2	224	2	249
5k	—	—	2	124
10k	1	224	1	249
25k	1	89	1	99
50k	0	179	1	49
100k	0	89	1	24
250k	0	35	0	39
500k	0	17	0	19
1M	0	8	0	9
2.5M	—	—	0	3
5M	—	—	0	1

Note: Settings with an error of 1% or less are recommended.

[Legend]

Blank: Setting is not possible.

—: Setting possible, but error will result.

Table 16.6 indicates the maximum bit rates in asynchronous mode when the baud rate generator is used. Tables 16.7 to 16.9 list the maximum rates for internal clock output and external clock input.

Table 16.6 Maximum Bit Rates for Various Frequencies with Baud Rate Generator (Asynchronous Mode)

		Settings	
Pϕ (MHz)	Maximum Bit Rate (bits/s)	n	N
16	1000000	0	0
18	1125000	0	0
20	1250000	0	0
32	2000000	0	0
36	2250000	0	0
40	2500000	0	0

Table 16.7 Maximum Bit Rates with External Clock Input (Clock Synchronous Mode: t_{scyc} = 12_{cyc})

Pϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bits/s) (CKE1 = 1: External Clock Input)
16	1.3333	1333333.3
18	1.5000	1500000.0
20	1.6667	1666666.7

Table 16.8 Maximum Bit Rates with External Clock Input (Clock Synchronous Mode: t_{scyc} = 16_{cyc})

Pϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bits/s) (CKE1 = 1: External Clock Input)
32	2.0000	2000000.0
36	2.2500	2250000.0
40	2.5000	2500000.0

Table 16.9 Maximum Bit Rates with Internal Clock Output (Clock Synchronous Mode: $t_{scyc} = 8_{cyc}$)

Pϕ (MHz)	Maximum Bit Rate (bits/s) (CKE1 = 0: Internal Clock Output)
16	2000000.0
18	2250000.0
20	2500000.0
32	4000000.0
36	4500000.0
40	5000000.0

Note: In the clock synchronous mode, the maximum bit rates for internal clock output and external clock input can be different.

16.4 Operation

16.4.1 Overview

For serial communications, the SCI has an asynchronous mode in which characters are synchronized individually, and a clock synchronous mode in which communication is synchronized with clock pulses.

Asynchronous or clock synchronous mode is selected and the transmit format is specified in the serial mode register (SCSMR1) as shown in table 16.10. The SCI clock source is selected by the combination of the C/$\overline{\text{A}}$ bit in SCSMR1 and the CKE1 bit in the serial control register (SCSCR1) as shown in table 16.11.

(1) Asynchronous Mode

- Data length is selectable: 7 or 8 bits.
- Parity bit is selectable. So is the stop bit length (1 or 2 bits). The combination of the preceding selections constitutes the communication format and character length.
- In receiving, it is possible to detect framing errors, parity errors, overrun errors, and breaks.
- SCI clock source: The SCI operates on the clock supplied by the on-chip baud rate generator.

(2) Clock Synchronous Mode

- The transmission/reception format has a fixed 8-bit data length.
- In receiving, it is possible to detect overrun errors.
- An internal or external clock can be selected as the SCI clock source.
  - When an internal clock is selected, the SCI operates using the on-chip baud rate generator, and outputs a serial clock signal to external devices.
  - When an external clock is selected, the SCI operates on the input serial clock. The on-chip baud rate generator is not used.

Table 16.10 SCSMR1 Settings and SCI Communication Formats

SCSMR1 Settings					SCI Communication Format		
Bit 7 C/$\overline{\text{A}}$	**Bit 6 CHR**	**Bit 5 PE**	**Bit 3 STOP**	**Mode**	**Data Length**	**Parity Bit**	**Stop Bit Length**
0	0	0	0	Asynchronous	8-bit	Not set	1 bit
			1				2 bits
		1	0			Set	1 bit
			1				2 bits
	1	0	0		7-bit	Not set	1 bit
			1				2 bits
		1	0			Set	1 bit
			1				2 bits
1	x	x	x	Clock synchronous	8-bit	Not set	None

[Legend]

x: Don't care

Table 16.11 SCSMR1 and SCSCR1 Settings and SCI Clock Source Selection

SCSMR1 Setting	SCSCR1 Setting			
Bit 7 C/$\overline{\text{A}}$	**Bit 1 CKE1**	**Mode**	**Clock Source**	**SCK Pin Function**
0	0*	Asynchronous	Internal	SCI does not use the SCK pin.
1	0	Clock synchronous	Internal	Serial clock is output.
	1		External	Input the serial clock.

Note: * Do not set the CKE1 bit to 1 in asynchronous mode. Otherwise, correct operation cannot be guaranteed.

16.4.2 Operation in Asynchronous Mode

In asynchronous mode, each transmitted or received character begins with a start bit and ends with a stop bit. Serial communications are synchronized one character at a time.

The transmitting and receiving sections of the SCI are independent, so full duplex communications are possible. Both the transmitter and receiver have a double-buffered structure so that data can be read or written during transmission or reception, enabling continuous data transfer.

Figure 16.2 shows the general format of asynchronous serial communications. In asynchronous serial communications, the communication line is normally held in the mark (high) state. The SCI monitors the line and starts serial communications when the line goes to the space (low) state, indicating a start bit. One serial character consists of a start bit (low), data (LSB first), parity bit (high or low), and stop bit (high), in that order.

When receiving in asynchronous mode, the SCI synchronizes at the falling edge of the start bit. The SCI samples each data bit on the eighth pulse of a clock with a frequency 16 times the bit rate. Receive data is latched at the center of each bit.

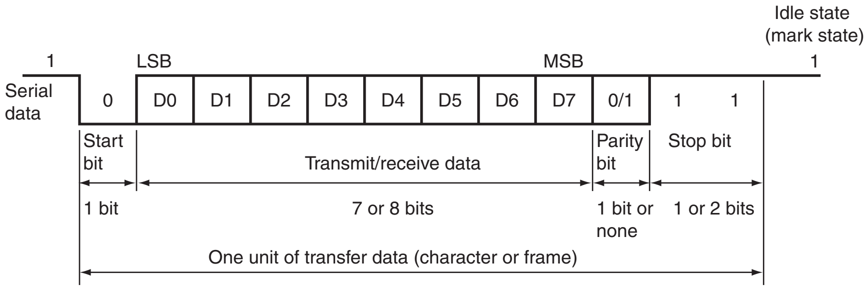


Figure 16.2 Example of Data Format in Asynchronous Communications (8-Bit Data with Parity and Two Stop Bits)

(1) Transmit/Receive Formats

Table 16.12 shows the transfer formats that can be selected in asynchronous mode. Any of eight transfer formats can be selected according to the SCSMR1 settings.

Table 16.12 Serial Transfer Formats (Asynchronous Mode)

<table>
<tr><th colspan="3">SCSMR1 Settings</th><th colspan="12">Serial Transfer Format and Frame Length</th></tr>
<tr><th>CHR</th><th>PE</th><th>STOP</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>S</td><td colspan="8">8-bit data</td><td>STOP</td><td></td><td></td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>S</td><td colspan="8">8-bit data</td><td>STOP</td><td>STOP</td><td></td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>S</td><td colspan="8">8-bit data</td><td>P</td><td>STOP</td><td></td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>S</td><td colspan="8">8-bit data</td><td>P</td><td>STOP</td><td>STOP</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>S</td><td colspan="7">7-bit data</td><td>STOP</td><td></td><td></td><td></td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>S</td><td colspan="7">7-bit data</td><td>STOP</td><td>STOP</td><td></td><td></td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>S</td><td colspan="7">7-bit data</td><td>P</td><td>STOP</td><td></td><td></td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>S</td><td colspan="7">7-bit data</td><td>P</td><td>STOP</td><td>STOP</td><td></td></tr>
</table>

[Legend]
S: Start bit
STOP: Stop bit
P: Parity bit

(2) Clock

An internal clock generated by the on-chip baud rate generator can be used as the SCI transmit/receive clock. The CKE1 bit in the serial control register (SCSCR1) must be set to 0 in asynchronous mode. Otherwise, correct operation cannot be guaranteed. For clock source selection, refer to table 16.11.

(3) Transmitting and Receiving Data

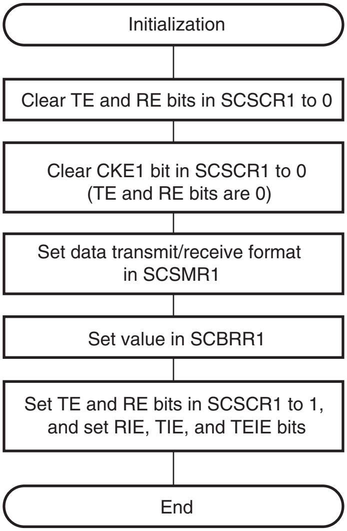
SCI Initialization (Asynchronous Mode):

Before transmitting or receiving, clear the TE and RE bits to 0 in the serial control register (SCSCR1), then initialize the SCI as follows.

If the TIE bit in SCSCR1 is set to 1 when the TE bit is 0, a transmit-data-empty interrupt (TXI) request is generated. To disable the TXI request when initializing the SCI, the TE and TIE bits must be cleared to 0 simultaneously. If the RIE bit in SCSCR1 and the RDRF flag are set to 1 when the RE bit is 0, a receive-data-full interrupt (RXI) request is generated. To disable the RXI request when initializing the SCI, the RE and RIE bits must be cleared to 0 simultaneously.

When changing the operation mode or the communication format, always clear the TE and RE bits to 0 before following the procedure given below. Clearing the TE bit to 0 sets the TDRE flag to 1 and initializes the transmit shift register (SCTSR1). Clearing the RE bit to 0, however, does not initialize the RDRF, PER, FER, and ORER flags or receive data register (SCRDR1), which retain their previous contents.

When an external clock is used, the clock should not be stopped during initialization or subsequent operation. SCI operation becomes unreliable if the clock is stopped.


Initialization
Clear TE and RE bits in SCSCR1 to 0
Clear CKE1 bit in SCSCR1 to 0
(TE and RE bits are 0)
Set data transmit/receive format
in SCSMR1
Set value in SCBRR1
Set TE and RE bits in SCSCR1 to 1,
and set RIE, TIE, and TEIE bits
End

[1] Clear the CKE1 bit in SCSCR1 to 0. Be sure to clear the RIE, TIE, TEIE, TE, and RE bits to 0.

[2] Set the data transmit/receive format in SCSMR1.

[3] Write a value corresponding to the bit rate to SCBRR1.

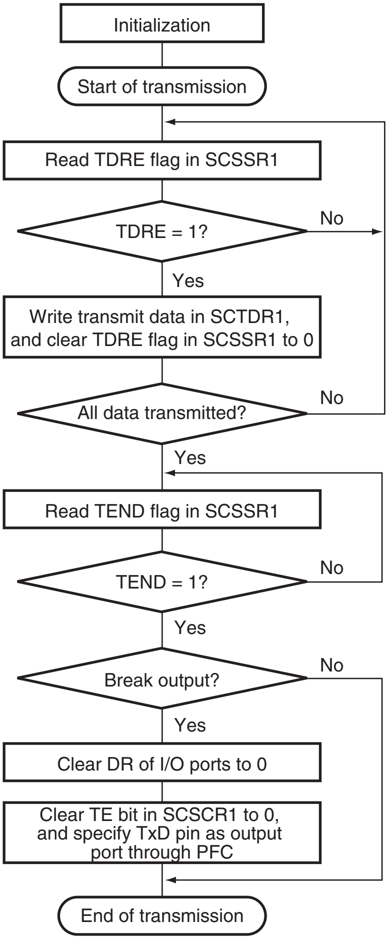
[4] Set the TE bit or RE bit in SCSCR1 to 1. Also set the RIE, TIE, and TEIE bits. Setting the TE and RE bits enables the TxD and RxD pins to be used. When transmitting, the mark state is entered; when receiving, the idle state is entered, waiting for a start bit.

Figure 16.3 Sample Flowchart for SCI Initialization

Transmitting Serial Data (Asynchronous Mode):

Figure 16.4 shows a sample flowchart for serial transmission.

Use the following procedure for serial data transmission after enabling the SCI for transmission.



[1] SCI initialization: Specify the TxD pin through the PFC.

[2] SCI status check and transmit data write: Read the serial status register (SCSSR1) and check that the TDRE flag is set to 1, write transmit data to the transmit data register (SCTDR1), and then clear the TDRE flag to 0.

[3] Serial transmission continuation procedure: To continue serial transmission, read 1 from the TDRE flag to confirm that writing is possible, then write data to SCTDR1, and then clear the TDRE flag to 0. Checking and clearing of the TDRE flag is automatic when the automotive direct memory access controller (A-DMAC) is activated by a transmit-data-empty interrupt (TXI) request, and data is written to SCTDR1.

[4] Break output at the end of serial transmission: To output a break in serial transmission, clear the data register (DR) of the I/O ports, clear the TE bit in the serial control register (SCSCR1) to 0, and specify the TxD pin as an output port through the PFC.

Figure 16.4 Sample Flowchart for Transmitting Serial Data

In serial transmission, the SCI operates as described below.

1. The SCI monitors the TDRE flag in the serial status register (SCSSR1). If it is cleared to 0, the SCI recognizes that data has been written to the transmit data register (SCTDR1) and transfers the data from SCTDR1 to the transmit shift register (SCTSR1).
2. After transferring data from SCTDR1 to SCTSR1, the SCI sets the TDRE flag to 1 and starts transmission. If the TIE bit in the serial control register (SCSCR1) is set to 1 at this time, a transmit-data-empty interrupt (TXI) request is generated.

   The serial transmit data is sent from the TxD pin in the following order.

   A. Start bit: One-bit 0 is output.

   B. Transmit data: 8-bit or 7-bit data is output in LSB-first order.

   C. Parity bit: One parity bit (even or odd parity) is output. (A format in which parity bit is not output can also be selected.)

   D. Stop bit(s): One or two 1 bits (stop bits) are output.

   E. Mark state: 1 is output continuously until the start bit that starts the next transmission is sent.
3. The SCI checks the TDRE flag at the timing for sending the stop bit.

   If the TDRE flag is 0, the data is transferred from SCTDR1 to SCTSR1, the stop bit is sent, and then serial transmission of the next frame is started.

   If the TDRE flag is 1, the TEND flag in SCSSR1 is set to 1, the stop bit is sent, and then the "mark state" is entered in which 1 is output. If the TEIE bit in SCSCR1 is set to 1 at this time, a TEI interrupt request is generated.

Figure 16.5 shows an example of the operation for transmission.

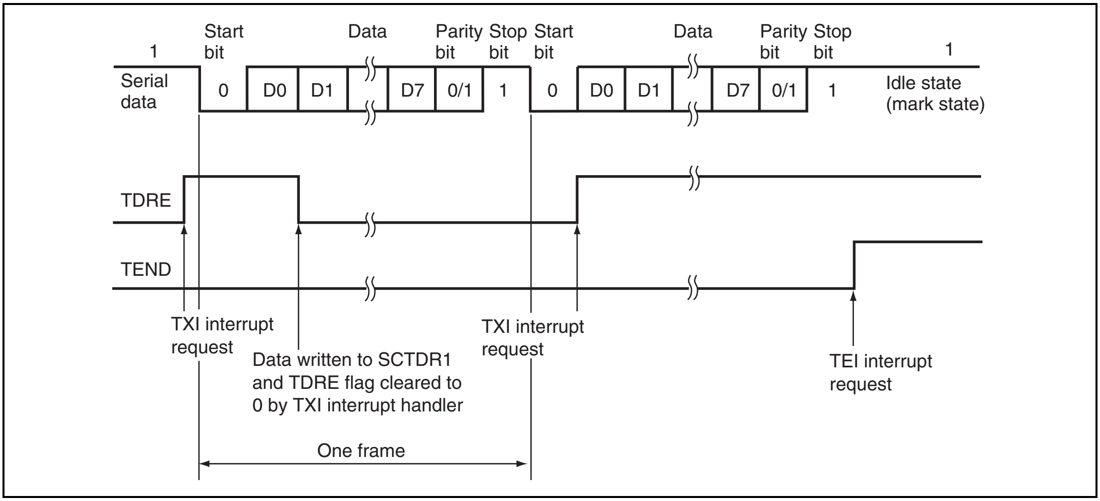
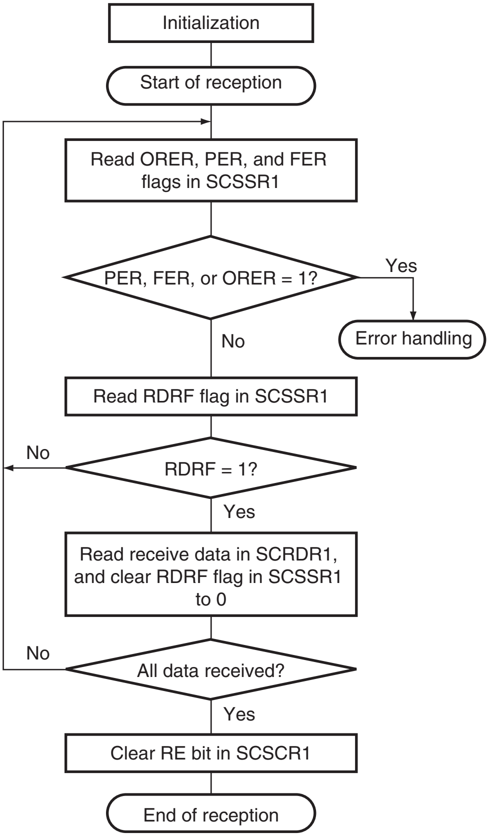


Figure 16.5 Example of Transmission in Asynchronous Mode (8-Bit Data, Parity, One Stop Bit)

Receiving Serial Data (Asynchronous Mode):

Figure 16.6 shows a sample flowchart for serial reception.

Use the following procedure for serial data reception after enabling the SCI for reception.



[1] SCI initialization: Specify the RxD pin through the PFC.

[2] Receive error handling and break detection: If a receive error occurs, read the ORER, PER, and FER flags in SCSSR1 to identify the error. After performing the appropriate error processing, ensure that the ORER, PER, and FER flags are all cleared to 0. Reception cannot be resumed if any of these flags are set to 1. In the case of a framing error, a break can also be detected by reading the value of the RxD pin.

[3] SCI status check and receive data read: Read the serial status register (SCSSR1) and check that RDRF = 1, then read the receive data in the receive data register (SCRDR1), and clear the RDRF flag to 0.

[4] Serial reception continuation procedure: To continue serial reception, clear the RDRF flag to 0 before the stop bit for the current frame is received. The RDRF flag is cleared automatically when the automotive direct memory access controller (A-DMAC) is activated by RXI to read the SCRDR1 value, thus this step is not needed.

Figure 16.6 Sample Flowchart for Receiving Serial Data (1)

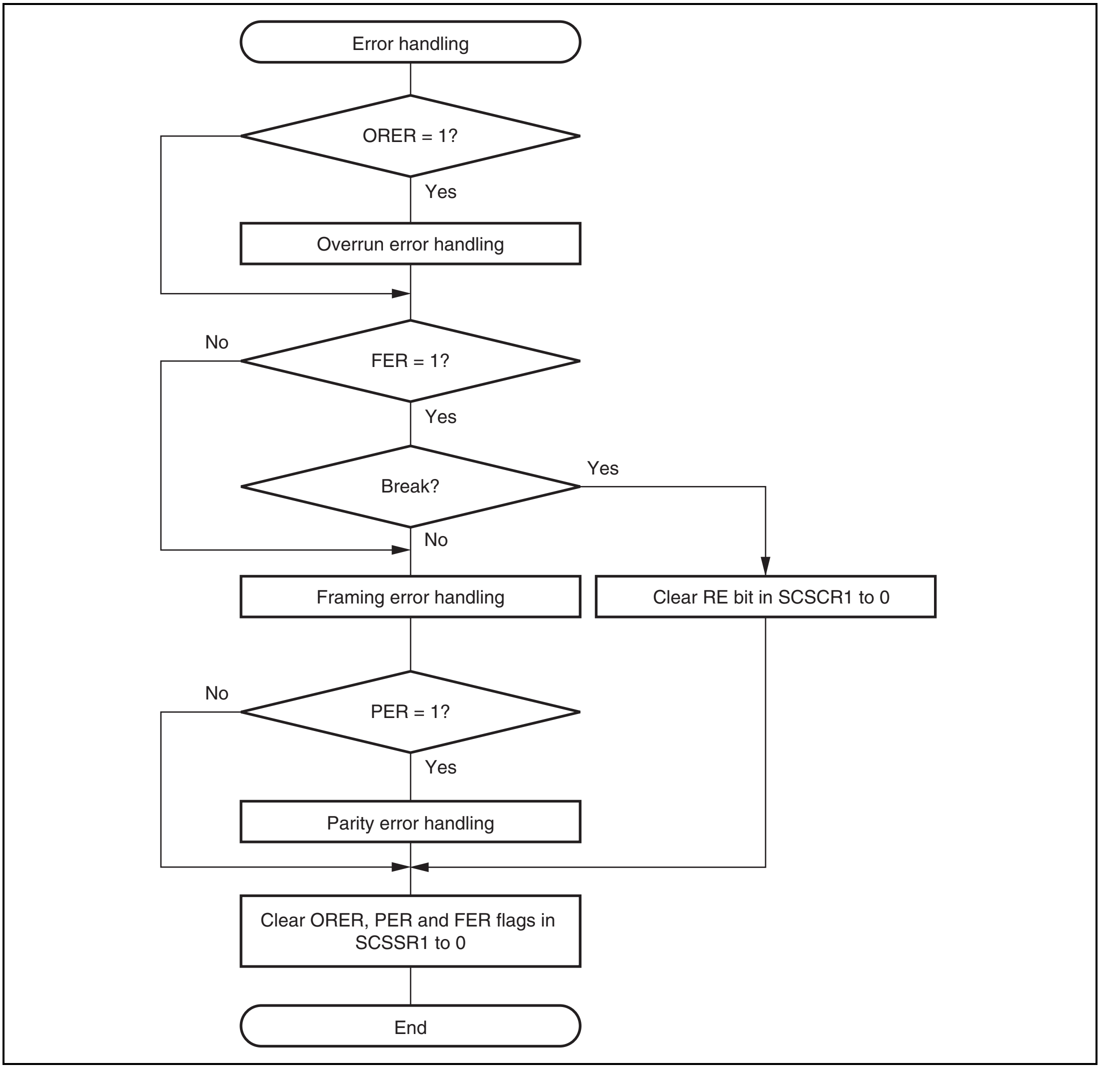


Figure 16.6 Sample Flowchart for Receiving Serial Data (2)

In serial reception, the SCI operates as described below.

1. The SCI monitors the transmission line, and if a 0 start bit is detected, performs internal synchronization and starts reception.
2. The received data is stored in SCRSR1 in LSB-to-MSB order.
3. The parity bit and stop bit are received.

   After receiving these bits, the SCI carries out the following checks.

   A. Parity check: The SCI counts the number of 1s in the received data and checks whether the count matches the even or odd parity specified by the $O/\overline{E}$ bit in the serial mode register (SCSMR1).

   B. Stop bit check: The SCI checks whether the stop bit is 1. If there are two stop bits, only the first is checked.

   C. Status check: The SCI checks whether the RDRF flag is 0 and the received data can be transferred from the receive shift register (SCRSR1) to SCRDR1.

   If all the above checks are passed, the RDRF flag is set to 1 and the received data is stored in SCRDR1. If a receive error is detected, the SCI operates as shown in table 16.13.

   Note: When a receive error occurs, subsequent reception cannot be continued. In addition, the RDRF flag will not be set to 1 after reception; be sure to clear the error flag to 0.

4. If the RIE bit in SCSCR1 is set to 1 when the RDRF flag changes to 1, a receive-data-full interrupt (RXI) request is generated. If the RIE bit in SCSCR1 is set to 1 when the ORER, PER, or FER flag changes to 1, a receive error interrupt (ERI) request is generated.

Table 16.13 Receive Errors and Error Conditions

Receive Error	Abbreviation	Error Condition	Data Transfer
Overrun error	ORER	When the next data reception is completed while the RDRF flag in SCSSR1 is set to 1	The received data is not transferred from SCRSR1 to SCRDR1.
Framing error	FER	When the stop bit is 0	The received data is transferred from SCRSR1 to SCRDR1.
Parity error	PER	When the received data does not match the even or odd parity specified in SCSMR1	The received data is transferred from SCRSR1 to SCRDR1.

Figure 16.7 shows an example of the operation for reception.

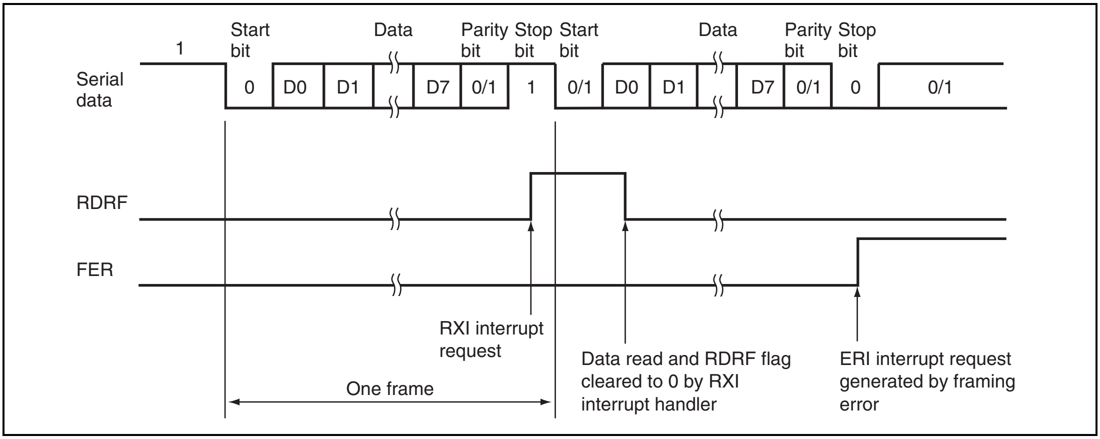


Figure 16.7 Example of SCI Receive Operation (8-Bit Data, Parity, One Stop Bit)

16.4.3 Clock Synchronous Mode

In clock synchronous mode, the SCI transmits and receives data in synchronization with clock pulses. This mode is suitable for high-speed serial communications.

The SCI transmitter and receiver are independent, so full-duplex communications are possible while sharing the same clock. Both the transmitter and receiver have a double-buffered structure so that data can be read or written during transmission or reception, enabling continuous data transfer.

Figure 16.8 shows the general format in clock synchronous serial communications.

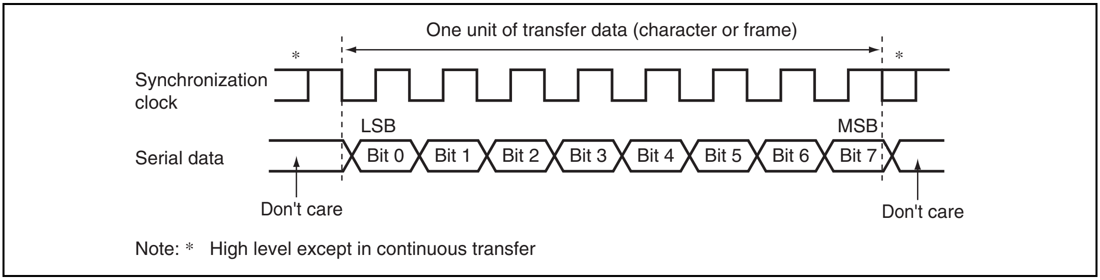


Figure 16.8 Data Format in Clock Synchronous Communications

In clock synchronous serial communications, each data bit is output on the communication line from one falling edge of the serial clock to the next. Data is guaranteed valid at the rising edge of the serial clock. In each character, the serial data bits are transmitted in order from the LSB (first) to the MSB (last). After output of the MSB, the communication line remains in the state of the MSB. In clock synchronous mode, the SCI transmits or receives data by synchronizing with the rising edge of the serial clock.

(1) Communication Format

The data length is fixed at eight bits. No parity bit can be added.

(2) Clock

An internal clock generated by the on-chip baud rate generator or an external clock input from the SCK pin can be selected as the SCI transmit/receive clock. The SCI clock source is selected by the combination of the $C/\overline{A}$ bit in SCSMR1 and the CKE1 bit in the serial control register (SCSCR1). For clock source selection, refer to table 16.11.

When the SCI operates on an internal clock, it outputs the clock signal at the SCK pin. Eight clock pulses are output per transmitted or received character. When the SCI is not transmitting or receiving, the clock signal remains in the high state.

(3) Transmitting and Receiving Data

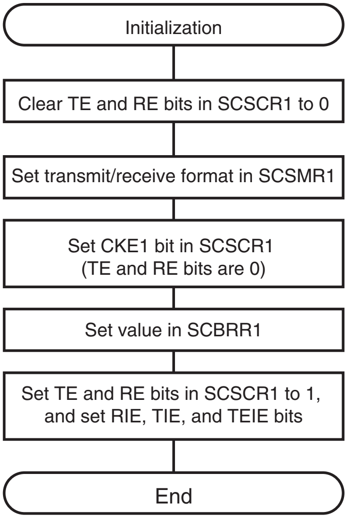
SCI Initialization (Clock Synchronous Mode):

Before transmitting or receiving, clear the TE and RE bits to 0 in the serial control register (SCSCR1), then initialize the SCI as follows.

If the TIE bit in SCSCR1 is set to 1 when the TE bit is 0, a transmit-data-empty interrupt (TXI) request is generated. To disable the TXI request when initializing the SCI, the TE and TIE bits must be cleared to 0 simultaneously. If the RIE bit in SCSCR1 and the RDRF flag are set to 1 when the RE bit is 0, a receive-data-full interrupt (RXI) request is generated. To disable the RXI request when initializing the SCI, the RE and RIE bits must be cleared to 0 simultaneously.

When changing the operation mode or the communication format, always clear the TE and RE bits to 0 before following the procedure given below. Clearing TE to 0 sets the TDRE flag to 1 and initializes the transmit shift register (SCTSR1). Clearing RE to 0, however, does not initialize the RDRF, PER, FER, and ORER flags and receive data register (SCRDR1), which retain their previous contents.

Figure 16.9 shows a sample flowchart for initializing the SCI.


Initialization
Clear TE and RE bits in SCSCR1 to 0
Set transmit/receive format in SCSMR1
Set CKE1 bit in SCSCR1 (TE and RE bits are 0)
Set value in SCBRR1
Set TE and RE bits in SCSCR1 to 1, and set RIE, TIE, and TEIE bits
End

[1] Clear the RIE, TIE, TEIE, TE, and RE bits in SCSCR1 to 0.

[2] Set the data transmit/receive format in SCSMR1.

[3] Select a clock through SCSCR1.

[4] Write a value corresponding to the bit rate to SCBRR1. Not necessary if an external clock is used.

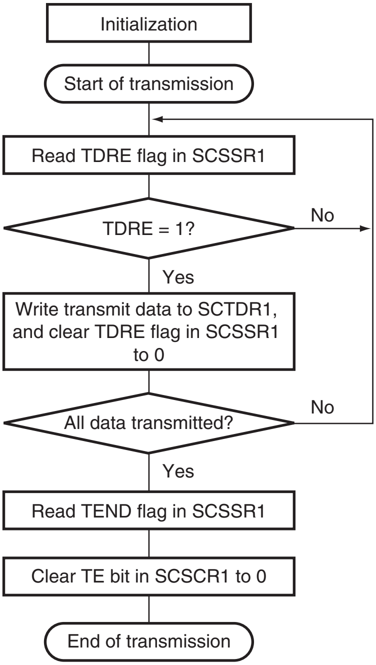
[5] Set the TE and RE bits in SCSCR1 to 1. Also set the RIE, TIE, and TEIE bits. Setting the TE and RE bits enables the TxD and RxD pins to be used.

Figure 16.9 Sample Flowchart for SCI Initialization

Transmitting Serial Data (Clock Synchronous Mode):

Figure 16.10 shows a sample flowchart for transmitting serial data.

Use the following procedure for serial data transmission after enabling the SCI for transmission.



[1] SCI initialization: Specify the TxD pin through the PFC.

[2] SCI status check and transmit data write: Read SCSSR1 and check that the TDRE flag is set to 1, then write transmit data to the transmit data register (SCTDR1), and clear the TDRE flag to 0.

[3] Serial transmission continuation procedure: To continue serial transmission, read 1 from the TDRE flag to confirm that writing is possible, write data to SCTDR1, and then clear the TDRE flag to 0. Checking and clearing of the TDRE flag is automatic when the automotive direct memory access controller (A-DMAC) is activated by a transmit-data-empty interrupt (TXI) request, and data is written to SCTDR1.

Figure 16.10 Sample Flowchart for Transmitting Serial Data

In transmitting serial data, the SCI operates as follows:

1. The SCI monitors the TDRE flag in the serial status register (SCSSR1). If it is cleared to 0, the SCI recognizes that data has been written to the transmit data register (SCTDR1) and transfers the data from SCTDR1 to the transmit shift register (SCTSR1).
2. After transferring data from SCTDR1 to SCTSR1, the SCI sets the TDRE flag to 1 and starts transmission. If the transmit-data-empty interrupt enable bit (TIE) in the serial control register (SCSCR1) is set to 1 at this time, a transmit-data-empty interrupt (TXI) request is generated.

   If clock output mode is selected, the SCI outputs eight synchronous clock pulses. If an external clock source is selected, the SCI outputs data in synchronization with the input clock. Data is output from the TxD pin in order from the LSB (bit 0) to the MSB (bit 7).
3. The SCI checks the TDRE flag at the timing for sending the MSB (bit 7). If the TDRE flag is 0, the data is transferred from SCTDR1 to SCTSR1 and serial transmission of the next frame is started, If the TDRE flag is 1, the TEND flag in SCSSR1 is set to 1, the MSB (bit 7) is sent, and then the TxD pin holds the states.

   If the TEIE bit in SCSCR1 is set to 1 at this time, a TEI interrupt request is generated.
4. After the end of serial transmission, the SCK pin is held in the high state.

Figure 16.11 shows an example of SCI transmit operation.

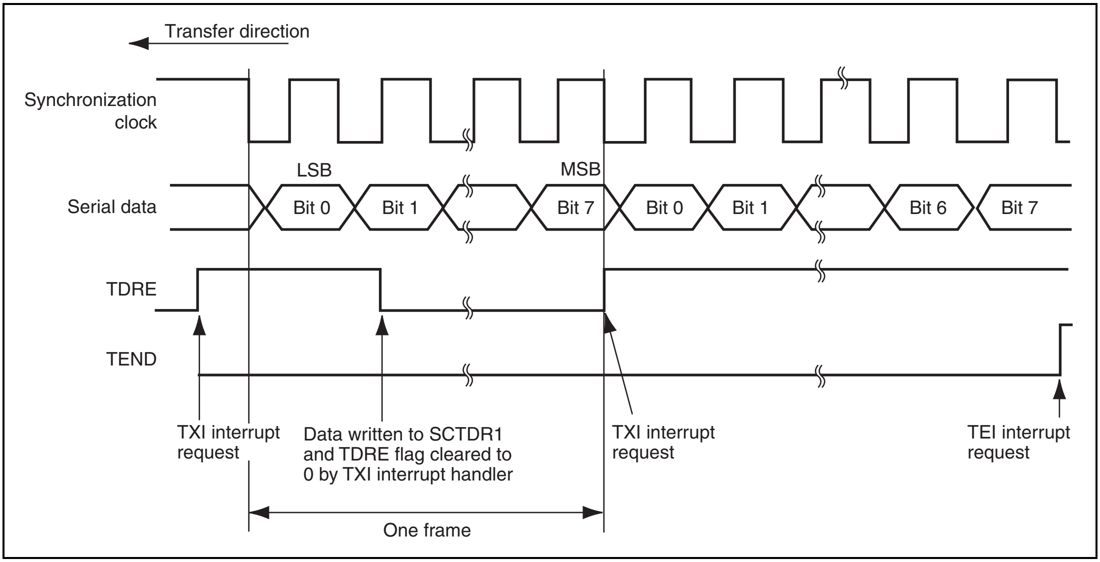
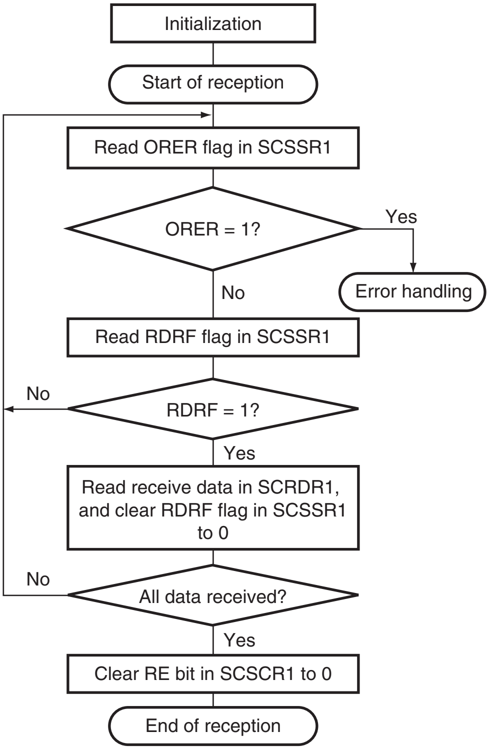


Figure 16.11 Example of SCI Transmit Operation

Receiving Serial Data (Clock Synchronous Mode):

Figure 16.12 shows a sample flowchart for receiving serial data. Use the following procedure for serial data reception after enabling the SCI for reception.

When switching from asynchronous mode to clock synchronous mode, make sure that the ORER, PER, and FER flags are all cleared to 0. If the FER or PER flag is set to 1, the RDRF flag will not be set and data reception cannot be started.



[1] SCI initialization: Specify the RxD pin through the PFC.

[2] Read the ORER flag in SCSSR1 to identify any error, perform the appropriate error handling, then clear the ORER flag to 0. Reception cannot be resumed while the ORER flag is set to 1.

[3] SCI status check and receive data read: Read SCSSR1 and check that the RDRF flag is set to 1, read the receive data in SCRDR1, and clear the RDRF flag to 0. The transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[4] Serial reception continuation procedure: To continue serial reception, read the receive data register (SCRDR1) and clear the RDRF flag to 0 before the MSB (bit 7) of the current frame is received. The RDRF flag is cleared automatically when the automotive direct memory access controller (A-DMAC) is activated by a receive-data-full interrupt (RXI) request to read the SCRDR1 value, thus this step is not needed.

Figure 16.12 Sample Flowchart for Receiving Serial Data (1)

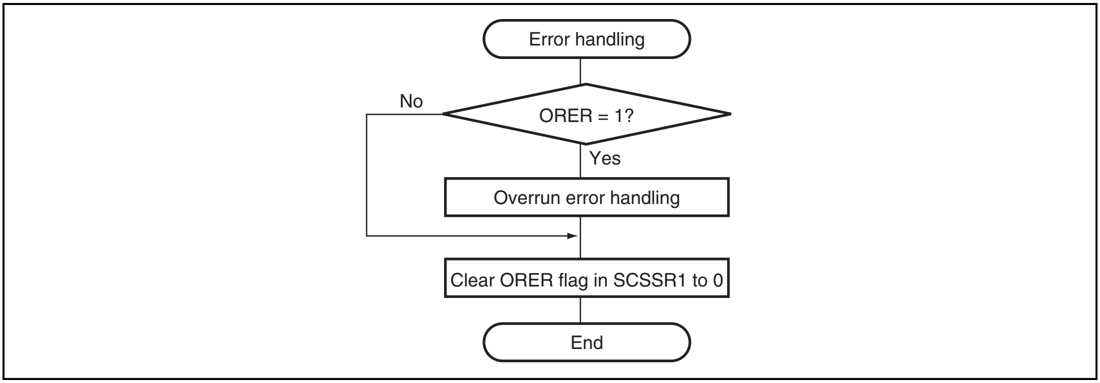


Figure 16.12 Sample Flowchart for Receiving Serial Data (2)

In receiving, the SCI operates as follows:

1. The SCI synchronizes with serial clock input or output and initializes internally.
2. Receive data is shifted into SCRSR1 in order from the LSB to the MSB. After receiving the data, the SCI checks whether the RDRF flag is 0 and the receive data can be transferred from SCRSR1 to SCRDR1. If this check is passed, the SCI sets the RDRF flag to 1 and stores the received data in SCRDR1. If a receive error is detected, the SCI operates as shown in table 16.13. In this state, subsequent reception cannot be continued. In addition, as the RDRF flag will be set to 1 after reception, be sure to clear the RDRF flag to 0.
3. After setting RDRF to 1, if the receive-data-full interrupt enable bit (RIE) is set to 1 in SCSCR1, the SCI requests a receive-data-full interrupt (RXI). If the ORER bit is set to 1 and the RIE bit in SCSCR1 is also set to 1, the SCI requests a receive error interrupt (ERI).

Figure 16.13 shows an example of SCI receive operation.

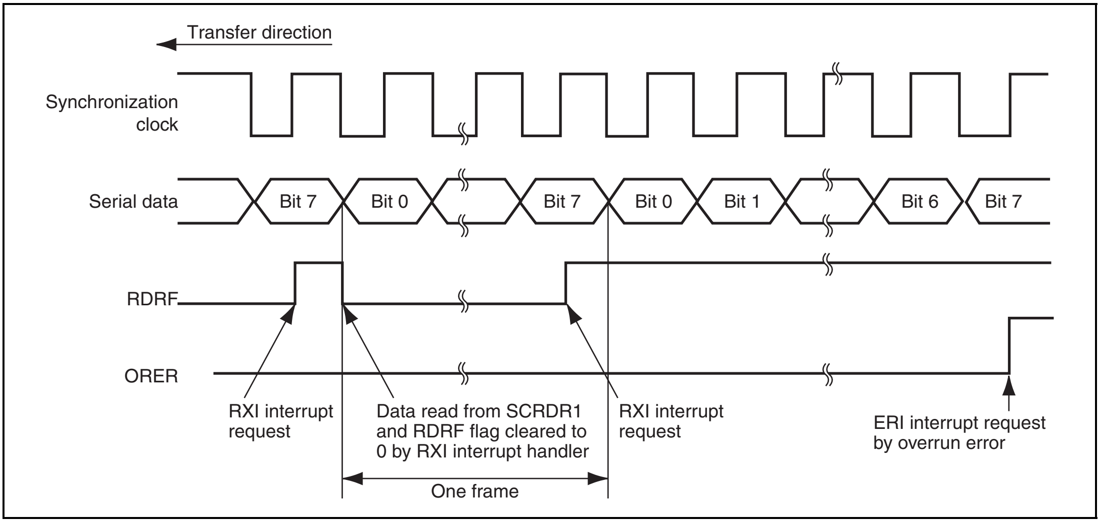
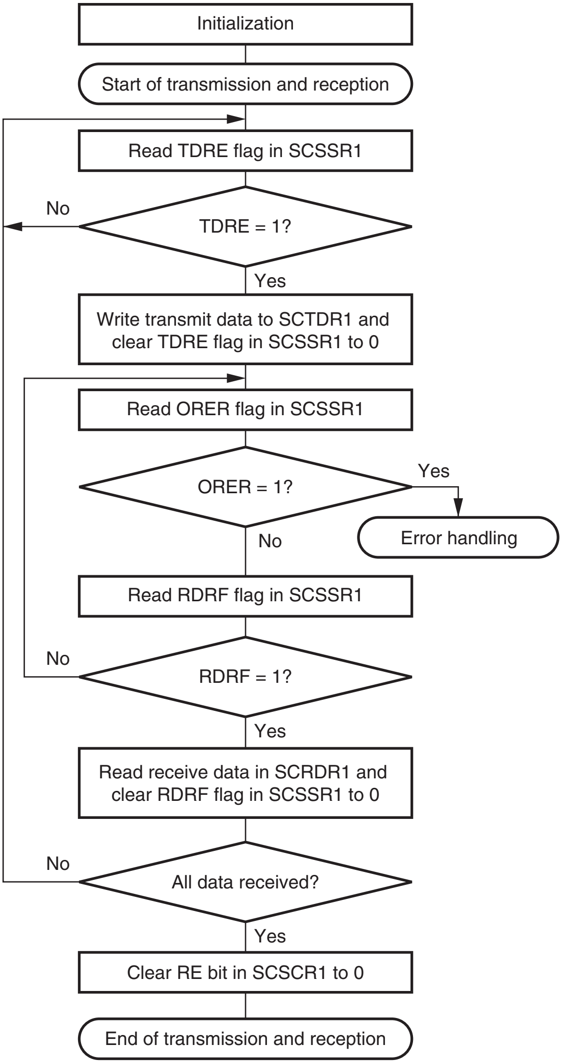


Figure 16.13 Example of SCI Receive Operation

Transmitting and Receiving Serial Data Simultaneously (Clock Synchronous Mode):

Figure 16.14 shows a sample flowchart for transmitting and receiving serial data simultaneously.

Use the following procedure for serial data transmission and reception after enabling the SCI for transmission and reception.


Initialization
Start of transmission and reception
Read TDRE flag in SCSSR1
No
TDRE = 1?
Yes
Write transmit data to SCTDR1 and clear TDRE flag in SCSSR1 to 0
Read ORER flag in SCSSR1
ORER = 1?
Yes
Error handling
No
Read RDRF flag in SCSSR1
No
RDRF = 1?
Yes
Read receive data in SCRDR1 and clear RDRF flag in SCSSR1 to 0
No
All data received?
Yes
Clear RE bit in SCSCR1 to 0
End of transmission and reception

[1] SCI initialization: Specify the TxD and RxD pins through the PFC.

[2] SCI status check and receive data write: Read SCSSR1 and check that the TDRE flag is set to 1, write the transmit data to SCTDR1, and then clear the TDRE flag to 0. Transition of the TDRE flag from 0 to 1 can also be identified by a TXI interrupt.

[3] Receive error handling: If a receive error occurs, read the ORER flag in SCSSR1, and after performing the appropriate error handling, clear the ORER flag to 0. Reception cannot be resumed if the ORER flag is set to 1.

[4] SCI status check and receive data read: Read SCSSR1 and check that the RDRF flag is set to 1, read the receive data in SCRDR1, and then clear the RDRF flag to 0. Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[5] Serial transmission/reception continuation procedure: To continue serial transmission/reception, before the MSB (bit 7) of the current frame is received, finish reading the RDRF flag, reading SCRDR1, and clearing the RDRF flag to 0. Also, before the MSB (bit 7) of the current frame is transmitted, read 1 from the TDRE flag to confirm that writing is possible. Then write data to SCTDR1 and clear the TDRE flag to 0. Checking and clearing of the TDRE flag is automatic when the automotive direct memory access controller (A-DMAC) is activated by a transmit-data-empty interrupt (TXI) request and data is written to SCTDR1. Also, the RDRF flag is cleared automatically when the A-DMAC is activated by a receive-data-full interrupt (RXI) request and the SCRDR1 value is read.

Note: When switching from transmit or receive operation to simultaneous transmit and receive operations, first clear the TE bit and RE bit to 0, then set both these bits to 1 simultaneously.

Figure 16.14 Sample Flowchart for Transmitting/Receiving Serial Data

16.5 SCI Interrupt Sources and A-DMAC

The SCI has four interrupt sources: transmit end (TEI), receive error (ERI), receive-data-full (RXI), and transmit-data-empty (TXI) interrupt requests. When DMA transfer to the A-DMAC is not enabled, these interrupt requests are sent to the interrupt controller separately. When DMA transfer is enabled, RXI and TXI requests are handled as DMA requests.

Table 16.14 shows the interrupt sources. The interrupt sources are enabled or disabled by means of the TIE, RIE, and TEIE bits in SCSCR1. DMA requests are enabled or disabled by the TIE and RIE bits in SCSCR1.

If the TDRE flag in the serial status register (SCSSR1) is set to 1 when the TIE bit in SCSCR1 is 1, a transmit-data-empty interrupt (TXI) request is generated. This request can be used to activate the automotive direct memory access controller (A-DMAC) to transfer data. To disable the TXI request to the A-DMAC, clear the TIE bit to 0.

The TDRE flag is automatically cleared to 0 when data is written to the transmit data register (SCTDR1) through the A-DMAC.

If the RDRF flag in SCSSR1 is set to 1 when the RIE bit in SCSCR1 is 1, a receive-data-full interrupt (RXI) request is generated. This request can be used to activate the A-DMAC to transfer data. To disable the RXI request to the A-DMAC, clear the RIE bit to 0.

The RDRF flag is automatically cleared to 0 when data is read from the receive data register (SCRDR1) through the A-DMAC.

When the ORER, FER, or PER flag in SCSSR1 is set to 1, an ERI interrupt request is generated. This request cannot be used to activate the A-DMAC. When processing the received data through the A-DMAC and handling the receive error by an interrupt requested to the CPU, set the RIE bit to 1.

When the TEND flag in SCSSR1 is set to 1, a TEI interrupt request is generated. This request cannot be used to activate the A-DMAC.

The TXI interrupt indicates that transmit data can be written, and the TEI interrupt indicates that transmission has been completed.

Table 16.14 SCI Interrupt Sources

Interrupt Source	Description	A-DMAC Activation	Priority on Reset Release
ERI	Interrupt caused by receive error (ORER, FER, or PER)	Not possible	High ↑
RXI	Interrupt caused by receive data full (RDRF)	Possible	
TXI	Interrupt caused by transmit data empty (TDRE)	Possible	
TEI	Interrupt caused by transmit end (TENT)	Not possible	↓ Low

16.6 Usage Notes

16.6.1 SCTDR1 Writing and TDRE Flag

The TDRE flag in the serial status register (SCSSR1) is a status flag indicating transferring of transmit data from SCTDR1 into SCTSR1. The SCI sets the TDRE flag to 1 when it transfers data from SCTDR1 to SCTSR1.

Data can be written to SCTDR1 regardless of the TDRE bit status.

If new data is written in SCTDR1 when TDRE is 0, however, the old data stored in SCTDR1 will be lost because the data has not yet been transferred to SCTSR1. Before writing transmit data to SCTDR1, be sure to check that the TDRE flag is set to 1.

16.6.2 Multiple Receive Error Occurrence

If multiple receive errors occur at the same time, the status flags in SCSSR1 are set as shown in table 16.15. When an overrun error occurs, data is not transferred from the receive shift register (SCRSR1) to the receive data register (SCRDR1) and the received data will be lost.

Table 16.15 SCSSR1 Status Flag Values and Transfer of Received Data

Receive Errors Generated	SCSSR1 Status Flags				Receive Data Transfer from SCRSR1 to SCRDR1
	RDRF	ORER	FER	PER	
Overrun error	1	1	0	0	Not transferred
Framing error	0	0	1	0	Transferred
Parity error	0	0	0	1	Transferred
Overrun error + framing error	1	1	1	0	Not transferred
Overrun error + parity error	1	1	0	1	Not transferred
Framing error + parity error	0	0	1	1	Transferred
Overrun error + framing error + parity error	1	1	1	1	Not transferred

16.6.3 Break Detection and Processing

Break signals can be detected by reading the RxD pin directly when a framing error (FER) is detected. In the break state the input from the RxD pin consists of all 0s, so the FER flag is set and the parity error flag (PER) may also be set. Note that, although transfer of receive data to SCRDR1 is halted in the break state, the SCI receiver continues to operate.

16.6.4 Sending a Break Signal

The TxD pin is used as a general I/O pin. The I/O condition and level of the TxD pin are determined by the data register (DR) of the I/O ports and the control register of the pin function controller (PFC). This feature can be used to send a break signal.

Until the TE bit is set to 1 (enabling transmission) after initialization of serial transmission, the TxD pin does not work. Until PFC settings complete, mark state is achieved using DR. Therefore, the TxD pin should be specified as an output port to output 1 initially. To send a break signal during serial transmission, clear DR to 0, then specify the TxD pin as an output port. When the TE bit is cleared to 0, the transmitter is initialized regardless of the current transmission state.

16.6.5 Receive Data Sampling Timing and Receive Margin (Asynchronous Mode)

The SCI operates on a base clock with a frequency of 16 times the transfer rate in asynchronous mode. In reception, the SCI synchronizes internally with the fall of the start bit, which it samples on the base clock. Receive data is latched at the rising edge of the eighth base clock pulse. The timing is shown in figure 16.15.

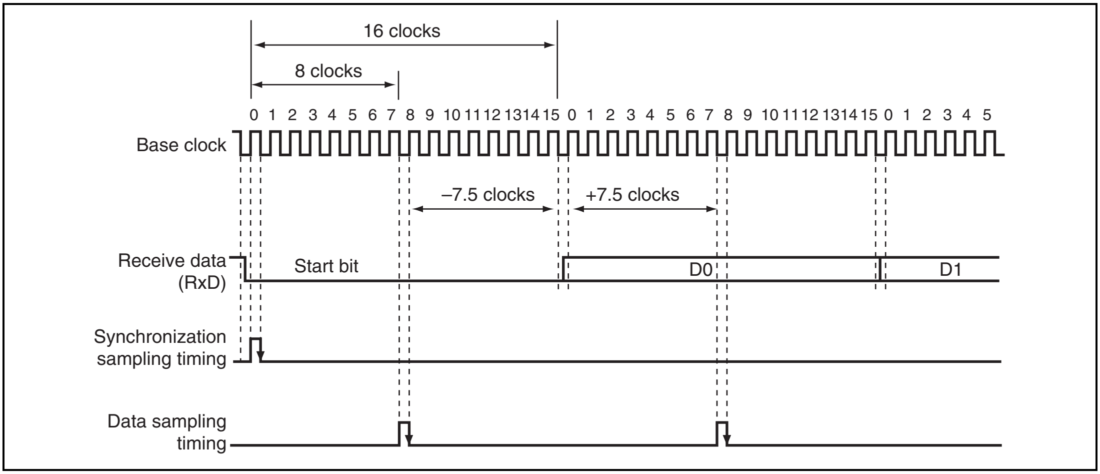


Figure 16.15 Receive Data Sampling Timing in Asynchronous Mode

The receive margin in asynchronous mode can therefore be expressed as shown in equation 1.

Equation 1:

$$M = \left| (0.5 - \frac{1}{2N}) - (L - 0.5)\,F - \frac{|D - 0.5|}{N}(1+F) \right| \times 100\ \%$$

M: Receive margin (%)
N: Ratio of bit rate to clock (N = 16)
D: Clock duty (D = 0 to 1.0)
L: Frame length (L = 9 to 12)
F: Absolute deviation of clock frequency

From equation 1, if F = 0 and D = 0.5, the receive margin is 46.875%, as given by equation 2.

Equation 2:

When D = 0.5 and F = 0:

$$M = (0.5 - 1/(2 \times 16)) \times 100\%$$

$$= 46.875\%$$

This is a theoretical value. A reasonable margin to allow in system designs is 20% to 30%.

16.6.6 Note on Using A-DMAC

When data is written to SCTDR1 by activating the A-DMAC by a TXI interrupt, the transmit end (TEND) flag value becomes undefined. In this case, do not use the TEND flag as the transmit end flag.

When the external clock source is used for the clock for synchronization, input the external clock after waiting for five or more Pϕ cycles after SCTDR1 is modified through the A-DMAC. If a transmit clock is input within four cycles after SCTDR1 is modified, a malfunction may occur (figure 16.16).

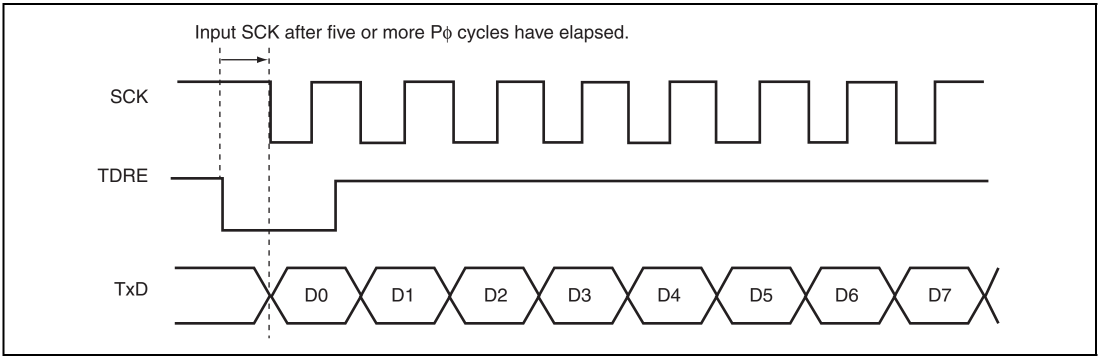


Figure 16.16 Example of Clock Synchronous Transfer Using A-DMAC

16.6.7 Note on Using External Clock in Clock Synchronous Mode

TE and RE must be set to 1 after waiting for four or more Pϕ cycles after the SCK external clock is changed from 0 to 1.

TE and RE must be set to 1 only while the SCK external clock is 1.

16.6.8 Note on Using A-DMAC

When the A-DMAC is used for transmitting data, the TIE bit in SCSCR1 should be set to 1 to enable a transmit-data-empty interrupt (TXI) request. When the A-DMAC is used for receiving data, the RIE bit in SCSCR1 should be set to 1 to enable a receive-data-full interrupt (RXI) request.

16.6.9 Serial Ports

Note that when the SCI pins are read through the serial ports, the value two Pϕ cycles before is read.

16.6.10 Note on Reception Only, with SCK Output, in Clock Synchronous Mode

In clock synchronous mode, only in the communications configured as reception only, with SCK pin used for output, the maximum baud rate is 1.25 Mbps.

Section 17 Renesas Serial Peripheral Interface (RSPI)

This LSI includes three-channel Renesas Serial Peripheral Interfaces (RSPI).

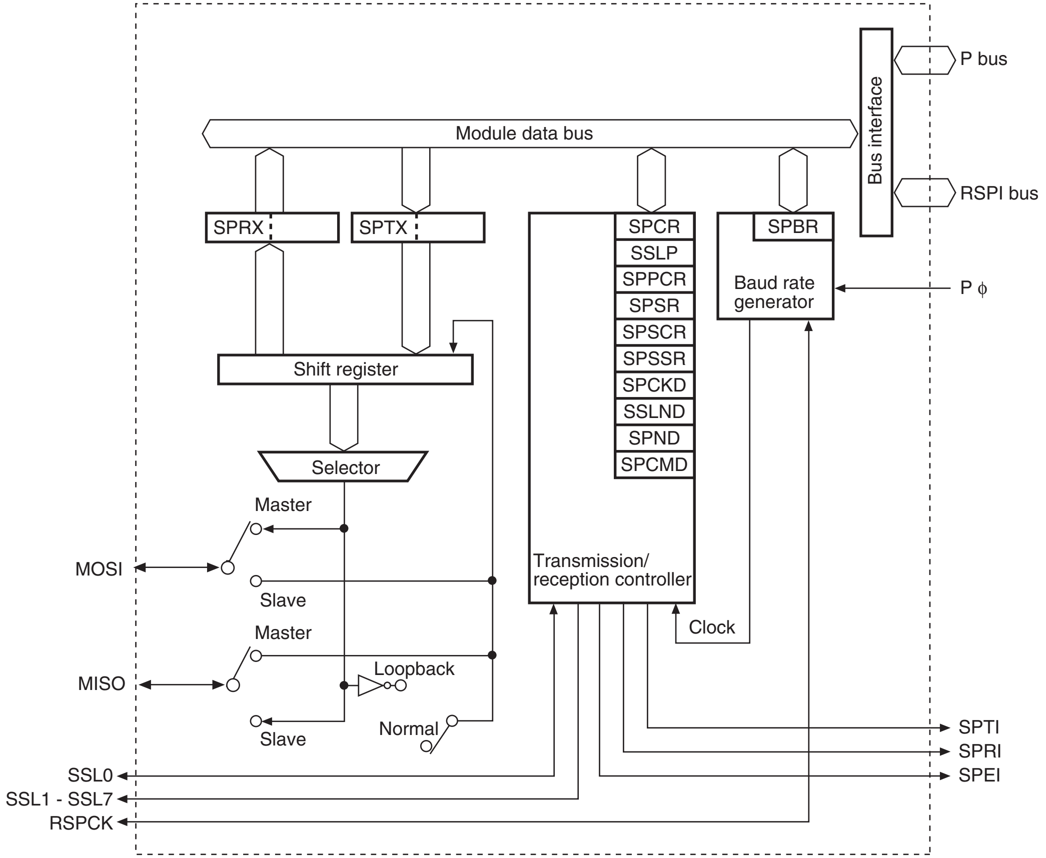
The RSPI has three channels which are independent of each other and is capable of full-duplex high-speed serial communications with multiple processors and peripheral devices.

17.1 Features

- The RSPI is a four-wire serial interface with the following configuration of signal lines.
  RSPCK (RSPI clock)
  SSL (slave select)
  MOSI (master output/slave in)
  MISO (master in/slave out)
- Capable of multi-master, single-master, and slave mode serial communications.
- Bit rate
  In master mode: 10.0 MHz at maximum when Pϕ = 20/40 MHz
  An internal baud rate generator generates RSPCK by dividing Pϕ (division by 2 to 4096)
  In slave mode: 2.5 MHz at maximum when Pϕ 20/40 = MHz
  Uses an external input clock as the serial clock (divided by 8 when one-time multiplication has been set for the peripheral clock; division by 16 when two-times multiplication has been set for the peripheral clock).
- Send/receive buffers in a double-buffer configuration.
- Data format
  Switchable MSB first/LSB first.
  Transfer bit length changeable to 8-16 bits.
- Modifiable RSPCK polarity/phase.
- SSL control function
  Modifiable SSL0 to SSL7 polarity.
  In single-master mode, outputs SSL0 to SSL7 signals.
  In multi-master mode, SSL0 signal for input, and SSL1 to SSL7 signals for either output or Hi-Z.
  In slave mode, SSL0 signal for input, and SSL1 to SSL7 signals for Hi-Z.
- In master mode, MOSI signal values can be set during SSL negation.
- Switchable CMOS output and open drain output.

- Serial transmission can be executed in sequential loops.
  Loops comprised of a maximum of eight commands.
  For each command, the following transfer formats can be set:
  - SSL0 to SSL7 signal output value
  - Data format
  - Burst transfer
  - A delay from SSL output assertion to RSPCK operation (RSPCK delay)
  - A delay from RSPCK stop to SSL output negation (SSL negation delay)
  - Wait for next-access SSL output (next-access delay)
  - RSPCK polarity and phase
  - Bit rate
- Receive buffer full flags provided (available as interrupts/DMA requests)
- Transmission buffer empty flags provided (available as interrupts/DMA requests)
- Mode default error flag provided (available as an interrupt request)
- Overrun error flag provided (available as an interrupt request)

Figure 17.1 shows an RSPI block diagram for one channel. When the CPU accesses the RSPI control registers, a peripheral bus (P-bus) is used. When A-DMAC accesses the RSPI data registers (SPDR), a special bus (RSPI bus) is used.



[Legend]

SPCR:	RSPI control register
SSLP:	RSPI slave select polarity register
SPPCR:	RSPI pin control register
SPSR:	RSPI status register
SPSCR:	RSPI sequence control register
SPSSR:	RSPI sequence status register
SPCKD:	RSPI clock delay register
SSLND:	RSPI slave select negate delay register
SPND:	RSPI next-access delay register
SPCMD:	RSPI command registers 0 to 7
SPBR:	RSPI bit rate register
SPTX:	RSPI transmission buffer (RSPI data register write side)
SPRX:	RSPI receive buffer (RSPI data register read side)
SPTI:	RSPI transmit interrupt
SPRI:	RSPI receive interrupt
SPEI:	RSPI error interrupt
RSPI Bus:	Data transfer bus between RSPI and A-DMAC

Figure 17.1 Block Diagram of RSPI (for One Channel)

17.2 Input/Output Pins

Table 17.1 shows the RSPI pin configuration. The RSPI automatically switches input/output directions of the pins. Pins SSLA0, SSLB0, and SSLC0 are set to outputs when the RSPI is a single master and are set to inputs when the RSPI is a multi master or a slave. Pins RSPCKA, MOSIA, MISOA, RSPCKB, MOSIB, MISOB, RSPCKC, MOSIC, and MISOC are set to inputs or outputs according to the following settings: whether the RSPI functions as a master or a slave and the input levels of pins SSLA0, SSLB0, and SSLC0 (see section 17.4.2, Controlling RSPI Pins).

Table 17.1 Pin Configuration

Channel	Pin Name	Symbol	I/O	Function
A	RSPI clock pin	RSPCKA	I/O	RSPI_A clock input/output
	Master transmit data pin	MOSIA	I/O	RSPI_A master transmit data
	Slave transmit data pin	MISOA	I/O	RSPI_A slave transmit data
	Slave select 0 pin	SSLA0	I/O	RSPI_A slave select
	Slave select 1 pin	SSLA1	Output	RSPI_A slave select
	Slave select 2 pin	SSLA2	Output	RSPI_A slave select
	Slave select 3 pin	SSLA3	Output	RSPI_A slave select
	Slave select 4 pin	SSLA4	Output	RSPI_A slave select
	Slave select 5 pin	SSLA5	Output	RSPI_A slave select
	Slave select 6 pin	SSLA6	Output	RSPI_A slave select
	Slave select 7 pin	SSLA7	Output	RSPI_A slave select
B	RSPI clock pin	RSPCKB	I/O	RSPI_B clock input/output
	Master transmit data pin	MOSIB	I/O	RSPI_B master transmit data
	Slave transmit data pin	MISOB	I/O	RSPI_B slave transmit data
	Slave select 0 pin	SSLB0	I/O	RSPI_B slave select
	Slave select 1 pin	SSLB1	Output	RSPI_B slave select
	Slave select 2 pin	SSLB2	Output	RSPI_B slave select
	Slave select 3 pin	SSLB3	Output	RSPI_B slave select

Channel	Pin Name	Symbol	I/O	Function
C	RSPI clock pin	RSPCKC	I/O	RSPI_C clock input/output
	Master transmit data pin	MOSIC	I/O	RSPI_C master transmit data
	Slave transmit data pin	MISOC	I/O	RSPI_C slave transmit data
	Slave select 0 pin	SSLC0	I/O	RSPI_C slave select
	Slave select 1 pin	SSLC1	Output	RSPI_C slave select
	Slave select 2 pin	SSLC2	Output	RSPI_C slave select
	Slave select 3 pin	SSLC3	Output	RSPI_C slave select

Note: Pin names RSPCK, MOSI, MISO, and SSL0 to SSL7 are used in the description for all channels, omitting the channel designation.

17.3 Register Descriptions

The RSPI has the registers shown in table 17.2. These registers enable the RSPI to perform the following controls: specifying master/slave modes, specifying a transfer format, and controlling the transmitter and receiver.

Table 17.2 Register Configuration

Chan-nel	Register Name	Symbol*[1]	R/W	Initial Value	Address	Access Size
A	RSPI control register A	SPCRA	R/W	H'00	H'FFFFB000	8, 16
	RSPI slave select polarity register A	SSLPA	R/W	H'00	H'FFFFB001	8
	RSPI pin control register A	SPPCRA	R/W	H'00	H'FFFFB002	8, 16
	RSPI status register A	SPSRA	R/(W)*[2]	H'20	H'FFFFB003	8
	RSPI data register A	SPDRA	R/W	H'0000	H'FFFFB004	16
	RSPI sequence control register A	SPSCRA	R/W	H'00	H'FFFFB008	8, 16
	RSPI sequence status register A	SPSSRA	R	H'00	H'FFFFB009	8
	RSPI bit rate register A	SPBRA	R/W	H'FF	H'FFFFB00A	8
	RSPI clock delay register A	SPCKDA	R/W	H'00	H'FFFFB00C	8, 16
	RSPI slave select negation delay register A	SSLNDA	R/W	H'00	H'FFFFB00D	8
	RSPI next-access delay register A	SPNDA	R/W	H'00	H'FFFFB00E	8
	RSPI command register A0	SPCMD0	R/W	H'070D	H'FFFFB010	16
	RSPI command register A1	SPCMD1	R/W	H'070D	H'FFFFB012	16
	RSPI command register A2	SPCMD2	R/W	H'070D	H'FFFFB014	16
	RSPI command register A3	SPCMD3	R/W	H'070D	H'FFFFB016	16
	RSPI command register A4	SPCMD4	R/W	H'070D	H'FFFFB018	16
	RSPI command register A5	SPCMD5	R/W	H'070D	H'FFFFB01A	16
	RSPI command register A6	SPCMD6	R/W	H'070D	H'FFFFB01C	16
	RSPI command register A7	SPCMD7	R/W	H'070D	H'FFFFB01E	16

Chan- nel	Register Name	Symbol*[1]	R/W	Initial Value	Address	Access Size
B	RSPI control register B	SPCRB	R/W	H'00	H'FFFFB800	8, 16
	RSPI slave select polarity register B	SSLPB	R/W	H'00	H'FFFFB801	8
	RSPI pin control register B	SPPCRB	R/W	H'00	H'FFFFB802	8, 16
	RSPI status register B	SPSRB	R/(W)*[2]	H'20	H'FFFFB803	8
	RSPI data register B	SPDRB	R/W	H'0000	H'FFFFB804	16
	RSPI sequence control register B	SPSCRB	R/W	H'00	H'FFFFB808	8, 16
	RSPI sequence status register B	SPSSRB	R	H'00	H'FFFFB809	8
	RSPI bit rate register B	SPBRB	R/W	H'FF	H'FFFFB80A	8
	RSPI clock delay register B	SPCKDB	R/W	H'00	H'FFFFB80C	8, 16
	RSPI slave select negation delay register B	SSLNDB	R/W	H'00	H'FFFFB80D	8
	RSPI next-access delay register B	SPNDB	R/W	H'00	H'FFFFB80E	8
	RSPI command register B0	SPCMDB0	R/W	H'070D	H'FFFFB810	16
	RSPI command register B1	SPCMDB1	R/W	H'070D	H'FFFFB812	16
	RSPI command register B2	SPCMDB2	R/W	H'070D	H'FFFFB814	16
	RSPI command register B3	SPCMDB3	R/W	H'070D	H'FFFFB816	16
	RSPI command register B4	SPCMDB4	R/W	H'070D	H'FFFFB818	16
	RSPI command register B5	SPCMDB5	R/W	H'070D	H'FFFFB81A	16
	RSPI command register B6	SPCMDB6	R/W	H'070D	H'FFFFB81C	16
	RSPI command register B7	SPCMDB7	R/W	H'070D	H'FFFFB81E	16

Chan- nel	Register Name	Symbol*[1]	R/W	Initial Value	Address	Access Size
C	RSPI control register C	SPCRC	R/W	H'00	H'FFFFC000	8, 16
	RSPI slave select polarity register C	SSLPC	R/W	H'00	H'FFFFC001	8
	RSPI pin control register C	SPPCRC	R/W	H'00	H'FFFFC002	8, 16
	RSPI status register C	SPSRC	R/(W)*[2]	H'20	H'FFFFC003	8
	RSPI data register C	SPDRC	R/W	H'0000	H'FFFFC004	16
	RSPI sequence control register C	SPSCRC	R/W	H'00	H'FFFFC008	8, 16
	RSPI sequence status register C	SPSSRC	R	H'00	H'FFFFC009	8
	RSPI bit rate register C	SPBRC	R/W	H'FF	H'FFFFC00A	8
	RSPI clock delay register C	SPCKDC	R/W	H'00	H'FFFFC00C	8, 16
	RSPI slave select negation delay register C	SSLNDC	R/W	H'00	H'FFFFC00D	8
	RSPI next-access delay register C	SPNDC	R/W	H'00	H'FFFFC00E	8
	RSPI command register C0	SPCMDC0	R/W	H'070D	H'FFFFC010	16
	RSPI command register C1	SPCMDC1	R/W	H'070D	H'FFFFC012	16
	RSPI command register C2	SPCMDC2	R/W	H'070D	H'FFFFC014	16
	RSPI command register C3	SPCMDC3	R/W	H'070D	H'FFFFC016	16
	RSPI command register C4	SPCMDC4	R/W	H'070D	H'FFFFC018	16
	RSPI command register C5	SPCMDC5	R/W	H'070D	H'FFFFC01A	16
	RSPI command register C6	SPCMDC6	R/W	H'070D	H'FFFFC01C	16
	RSPI command register C7	SPCMDC7	R/W	H'070D	H'FFFFC01E	16

Notes: 1. Register names and symbols are used in the description for all channels, omitting the channel designation.

2. Only 0 can be written to clear the flag.

17.3.1 RSPI Control Register (SPCR)

SPCR sets the operating mode of the RSPI. SPCR can be read from or written to by the CPU. If the MSTR and MODFEN bits are changed while the RSPI function is enabled by setting the SPE bit to 1, subsequent operations cannot be guaranteed.

Bit:	7	6	5	4	3	2	1	0
	SPRIE	SPE	SPTIE	SPEIE	MSTR	MOD FEN	-	-
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	SPRIE	0	R/W	RSPI Receive Interrupt Enable If the RSPI has detected a receive buffer write after completion of a serial transfer and the SPRF bit in the RSPI status register (SPSR) is set to 1, this bit enables or disables the generation of an RSPI receive interrupt request. 0: Disables the generation of RSPI receive interrupt requests. 1: Enables the generation of RSPI receive interrupt requests.
6	SPE	0	R/W	RSPI Function Enable Setting this bit to 1 enables the RSPI function. When the MODF bit in the RSPI status register (SPSR) is 1, the SPE bit cannot be set to 1 (see section 17.4.7, Error Detection). Setting the SPE bit to 0 disables the RSPI function, and initializes a part of the module function (see section 17.4.8, Initializing RSPI). 0: Disables the RSPI function 1: Enables the RSPI function

Bit	Bit Name	Initial Value	R/W	Description
5	SPTIE	0	R/W	RSPI Transmit Interrupt Enable Enables or disables the generation of RSPI transmit interrupt requests when the RSPI detects transmission buffer empty and sets the SPTEF bit in the RSPI status register (SPSR) to 1. In the RSPI disabled (with the SPE bit 0) status, the SPTEF bit is 1. Therefore, note that setting the SPTIE bit to 1 when the RSPI is in the disabled status generates an RSPI transmit interrupt request. 0: Disables the generation of RSPI transmit interrupt requests. 1: Enables the generation of RSPI transmit interrupt requests.
4	SPEIE	0	R/W	RSPI Error Interrupt Enable Enables or disables the generation of RSPI error interrupt requests when the RSPI detects a mode fault error and sets the MODF bit in the RSPI status register (SPSR) to 1, or when the RSPI detects and sets the OVRF bit in SPSR to 1 (see section 17.4.7, Error Detection). 0: Disables the generation of RSPI error interrupt requests. 1: Enables the generation of RSPI error interrupt requests.
3	MSTR	0	R/W	RSPI Master/Slave Mode Select Selects master/slave mode of RSPI. According to MSTR bit settings, the RSPI determines the direction of pins RSPCK, MOSI, MISO, and SSL1 to SSL7. 0: Slave mode 1: Master mode
2	MODFEN	0	R/W	Mode Fault Error Detection Enable Enables or disables the detection of mode fault error (see section 17.4.7, Error Detection). In addition, the RSPI determines the input/output directions of the SSL0 pin based on combinations of the MODFEN and MSTR bits (see section 17.4.2, Controlling RSPI Pins). 0: Disables the detection of mode fault error 1: Enables the detection of mode fault error

Bit	Bit Name	Initial Value	R/W	Description
1, 0	—	All 0	R	Reserved The write value should always be 0. Otherwise, operation cannot be guaranteed.

17.3.2 RSPI Slave Select Polarity Register (SSLP)

SSLP sets the polarity of the SSL0 to SSL7 signals of the RSPI. SSLP can always be read from or written to by the CPU. If the contents of SSLP are changed by the CPU while the RSPI function is enabled by setting the SPE bit in the RSPI control register (SPCR) to 1, subsequent operations cannot be guaranteed.

Note that the SSL4 to SSL7 pins do not exist in channels B and C, therefore the polarity of the SSL pins in those channels cannot be switched by setting the SSL4P to SSL7P bits in the SSLPB and SSLPC registers.

Bit:	7	6	5	4	3	2	1	0
	SSL7P	SSL6P	SSL5P	SSL4P	SSL3P	SSL2P	SSL1P	SSL0P
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	SSL7P	0	R/W	SSL Signal Polarity Setting These bits set the polarity of the SSL signals. 0: SSLi signal 0-active 1: SSLi signal 1-active
6	SSL6P	0	R/W	
5	SSL5P	0	R/W	
4	SSL4P	0	R/W	
3	SSL3P	0	R/W	
2	SSL2P	0	R/W	
1	SSL1P	0	R/W	
0	SSL0P	0	R/W	

[Legend]

i = 0 to 7

17.3.3 RSPI Pin Control Register (SPPCR)

SPPCR sets the modes of the RSPI pins. SPPCR can be read from or written to by the CPU. If the contents of this register are changed by the CPU while the RSPI function is enabled by setting the SPE bit in the RSPI control register (SPCR) to 1, operation cannot be guaranteed.

Bit:	7	6	5	4	3	2	1	0
	-	-	MOIFE	MOIFV	-	SPOM	-	SPLP
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R/W	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved The write value should always be 0. Otherwise, operation cannot be guaranteed.
5	MOIFE	0	R/W	MOSI Idle Value Fixing Enable Fixes the MOSI output value when the RSPI in master mode is in an SSL negation period (including the SSL retention period during a burst transfer). When MOIFE is 0, the RSPI outputs the last data from the previous serial transfer during the SSL negation period. When MOIFE is 1, the RSPI outputs the fixed value set in the MOIFV bit to the MOSI bit. 0: MOSI output value equals final data from previous transfer 1: MOSI output value equals the value set in the MOIFV bit
4	MOIFV	0	R/W	MOSI Idle Fixed Value If the MOIFE bit is 1 in master mode, the RSPI, according to MOIFV bit settings, determines the MOSI signal value during the SSL negation period (including the SSL retention period during a burst transfer). 0: MOSI Idle fixed value equals 0 1: MOSI Idle fixed value equals 1
3	—	0	R	Reserved The write value should always be 0. Otherwise, operation cannot be guaranteed.

Bit	Bit Name	Initial Value	R/W	Description
2	SPOM	0	R/W	RSPI Output Pin Mode Sets the RSPI output pins to CMOS output/open drain output. 0: CMOS output 1: Open-drain output
1	—	0	R	Reserved The write value should always be 0. Otherwise, operation cannot be guaranteed.
0	SPLP	0	R/W	RSPI Loopback When the SPLP bit is set to 1, the RSPI shuts off the path between the MISO pin and the shift register, and between the MOSI pin and the shift register, and connects (reverses) the input path and the output path for the shift register. 0: Normal mode 1: Loopback mode

17.3.4 RSPI Status Register (SPSR)

SPSR indicates the operating status of the RSPI. SPSR can be read by the CPU. Writing 1 to the SPRF, SPTEF, MODF, and OVRF bits cannot be performed by the CPU. These bits can be cleared to 0 after they are read as 1.

Bit:	7	6	5	4	3	2	1	0
	SPRF	-	SPTEF	-	-	MODF	-	OVRF
Initial value:	0	0	1	0	0	0	0	0
R/W:	R/(W)*	R	R/(W)*	R	R	R/(W)*	R	R/(W)*

Note: * Only 0 can be written to this bit after reading it as 1 to clear the flag.

Bit	Bit Name	Initial Value	R/W	Description
7	SPRF	0	R/(W)*	RSPI Receive Buffer Full Flag Indicates the status of the receive buffer for the RSPI data register (SPDR). Upon completion of a serial transfer with the SPRF bit 0, the RSPI transfers the receive data from the shift register to SPDR, and sets this bit to 1. The SPRF bit is cleared to 0 under the following conditions: • The CPU reads SPSR when the SPRF bit is 1, and then the CPU writes a 0 to the SPRF bit. • The A-DMAC reads received data from the SPDR. • System power-on reset or standby • If a serial transfer ends while the SPRF bit is 1, the RSPI does not transfer the received data from the shift register to the SPDR. When the OVRF bit in SPSR is 1, the SPRF bit cannot be changed from 0 to 1 (see section 17.4.7, Error Detection). 0: No valid data in SPDR 1: Valid data found in SPDR
6	—	0	R	Reserved The write value should always be 0. Otherwise, operation cannot be guaranteed.

Bit	Bit Name	Initial Value	R/W	Description
5	SPTEF	1	R/(W)*	RSPI Transmission Buffer Empty Flag Indicates the status of the transmission buffer for the RSPI data register (SPDR). After the initialization of RSPI or after transmit data is transferred from the transmission buffer to the shift register, the RSPI sets the SPTEF bit to 1. The SPTEF bit is cleared to 0 under the following conditions. If the SPTEF bit is cleared and the shift register is empty, the data is copied from the transmission buffer to the shift register. • The CPU reads SPRF when the SPTEF bit is 1, and then the CPU writes 0 to the SPTEF bit. • The A-DMAC writes the transmit data to SPDR. • The CPU and the A-DMAC can write to SPDR only when the SPTEF bit is 1. If the CPU or the A-DMAC writes to the transmission buffer of SPDR when the SPTEF bit is 0, the data in the transmission buffer is not updated. 0: Data found in the transmission buffer 1: No data in the transmission buffer
4, 3	—	All 0	R	Reserved The write value should always be 0. Otherwise, operation cannot be guaranteed.

Bit	Bit Name	Initial Value	R/W	Description
2	MODF	0	R/(W)*	Mode Fault Error Flag Indicates the occurrence of a mode fault error. When the input level of the SSL0 pin changes to the active level while the MSTR bit in the RSPI control register (SPCR) is 1 and the MODFEN bit is 1 with the RSPI being in multi-master mode, the RSPI detects a mode fault error and sets the MODF bit to 1. Similarly, if the MODFEN bit is set to 1 when the MSTR bit is 0 and the RSPI is in slave mode, and the SSL0 pin is negated before the RSPCK cycle necessary for data transfer ends, the RSPI detects a mode fault error. The active level of the SSL0 signal is determined by the SSL0P bit in the RSPI slave select polarity register (SSLP). The MODF bit is cleared to 0 under the following conditions. • The CPU reads SPSR when the MODF bit is 1, and then writes 0 to the MODF bit. • Power-on reset or standby 0: No mode fault error occurs 1: A mode fault error occurs
1	—	0	R	Reserved The write value should always be 0. Otherwise, operation cannot be guaranteed.
0	OVRF	0	R/(W)*	Overrun Error Flag Indicates the occurrence of an overrun error. If a serial transfer ends while the SPRF bit is 1, the RSPI detects an overrun error, and sets the OVRF bit to 1. The OVRF bit is cleared to 0 under the following conditions. • The CPU reads SPSR when the OVRF bit is 1, and then writes 0 to the OVRF bit. • Power-on reset or standby 0: No overrun error occurs 1: An overrun error occurs

Note: * Only 0 can be written to this bit after reading it as 1 to clear the flag.

17.3.5 RSPI Data Register (SPDR)

SPDR is a buffer that stores RSPI transmit/receive data. The transmit and receive buffers are independent registers each other and are allocated to the same address.

When the CPU or A-DMAC requests writing to SPDR and if the SPTEF bit in the RSPI status register (SPSR) is 1, the RSPI writes data to the transmission buffer of SPDR. If the SPTEF bit is 0, the RSPI does not update the SPDR transmission buffer.

SPDR receive buffer can be read by the CPU and A-DMAC. In the normal operating method, the CPU and A-DMAC read the receive buffer when the SPRF bit in SPSR is 1 (a condition in which unread data is stored in the receive buffer). When the SPRF bit or the OVRF bit in SPSR is 1, the RSPI does not update the receive buffer of SPSR at the termination of a serial transfer.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPD15	SPD14	SPD13	SPD12	SPD11	SPD10	SPD9	SPD8	SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

17.3.6 RSPI Sequence Control Register (SPSCR)

SPSCR sets the sequence controlled method when the RSPI operates in master mode. SPSCR can be read from or written to by the CPU. If the contents of SPSCR are changed by the CPU while the MSTR and SPE bits in the RSPI control register (SPCR) are 1 with the RSPI function enabled, the subsequent operation cannot be guaranteed.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	SPSLN[2:0]		
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

<table>
<tr><th>Bit</th><th>Bit Name</th><th>Initial Value</th><th>R/W</th><th>Description</th></tr>
<tr><td>7 to 3</td><td>—</td><td>All 0</td><td>R</td><td>Reserved
The write value should always be 0. Otherwise, operation cannot be guaranteed.</td></tr>
<tr><td>2 to 0</td><td>SPSLN[2:0]</td><td>000</td><td>R/W</td><td>RSPI Sequence Length Setting
These bits set a sequence length when the RSPI in master mode performs sequential operations. The RSPI in master mode changes RSPI command registers 0 to 7 (SPCMD0 to SPCMD7) to be referenced and the order in which they are referenced according to the sequence length that is set in the SPSLN2 to SPSLN0 bits. When the RSPI is in slave mode, SPCMD0 is always referenced.
The relationship among the setting in these bits, sequence length, and referenced SPCMD register number is shown below.
<table>
<tr><th>SPSLN [2:0]</th><th>Sequence Length</th><th>Referenced SPCMD #</th></tr>
<tr><td>000</td><td>1</td><td>0 → 0 → …</td></tr>
<tr><td>001</td><td>2</td><td>0 → 1 → 0 → …</td></tr>
<tr><td>010</td><td>3</td><td>0 → 1 → 2 → 0 → …</td></tr>
<tr><td>011</td><td>4</td><td>0 → 1 → 2 → 3 → 0 → …</td></tr>
<tr><td>100</td><td>5</td><td>0 → 1 → 2 → 3 → 4 → 0 → …</td></tr>
<tr><td>101</td><td>6</td><td>0 → 1 → 2 → 3 → 4 → 5 → 0 → …</td></tr>
<tr><td>110</td><td>7</td><td>0 → 1 → 2 → 3 → 4 → 5 → 6 → 0 → …</td></tr>
<tr><td>111</td><td>8</td><td>0 → 1 → 2 → 3 → 4 → 5 → 6 → 7 → 0 → …</td></tr>
</table></td></tr>
</table>

17.3.7 RSPI Sequence Status Register (SPSSR)

SPSSR indicates the sequence control status when the RSPI operates in master mode. SPSSR can be read by the CPU. Any writing to SPSSR by the CPU is ignored.

Bit:	7	6	5	4	3	2	1	0
	-	SPECM[2:0]			-	SPCP[2:0]		
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved The write value should always be 0. Otherwise, operation cannot be guaranteed.
6 to 4	SPECM[2:0]	000	R	RSPI Error Command These bits indicate RSPI command registers 0 to 7 (SPCMD0 to SPCMD7) that are pointed to by command pointers (SPCP2 to SPCP0 bits) when an error is detected during sequence control by the RSPI. The RSPI updates the bits SPECM2 to SPECM0 only when an error is detected. If both the OVRF and MODF bits in the RSPI status register (SPSR) are 0 and there is no error, the values of the bits SPECM2 to SPECM0 have no meaning. For the RSPI's error detection function, see section 17.4.7, Error Detection. For the RSPI's sequence control, see section 17.4.9, Master Mode Operation. 000: SPCMD0 001: SPCMD1 010: SPCMD2 011: SPCMD3 100: SPCMD4 101: SPCMD5 110: SPCMD6 111: SPCMD7
3	—	0	R	Reserved The write value should always be 0. Otherwise, operation cannot be guaranteed.

Bit	Bit Name	Initial Value	R/W	Description
2 to 0	SPCP[2:0]	000	R	RSPI Command Pointer During RSPI sequence control, these bits indicate RSPI command registers 0 to 7 (SPCMD0 to SPCMD7), which are currently pointed to by the pointers. For the RSPI's sequence control, see section 17.4.9, Master Mode Operation. 000: SPCMD0 001: SPCMD1 010: SPCMD2 011: SPCMD3 100: SPCMD4 101: SPCMD5 110: SPCMD6 111: SPCMD7

17.3.8 RSPI Bit Rate Register (SPBR)

SPBR set the bit rate in master mode. SPBR can be read from or written to by the CPU. If the contents of SPBR are changed by the CPU while the MSTR and SPE bits in the RSPI control register (SPCR) are 1 with the RSPI function in master mode enabled, operation cannot be guaranteed.

Bit:	7	6	5	4	3	2	1	0
	SPR7	SPR6	SPR5	SPR4	SPR3	SPR2	SPR1	SPR0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The bit rate is determined by combinations of SPBR settings and the bit settings in the BRDV1 and BRDV0 bits in the RSPI command registers (SPCMD0 to SPCMD7). The equation for calculating the bit rate is given below. In the equation, n denotes an SPBR setting (0, 1, 2, …, 255), and N denotes bit settings in the bits BRDV1 and BRDV0 (0, 1, 2, 3). The maximum bit rate is 10.0 Mbps in master mode. Normal operation cannot be guaranteed if the bit rate is set to exceed 10.0 Mbps.

$$\text{Bit rate} = \frac{f\,(P\phi)}{2 \times (n + 1) \times 2^{N}}$$

Table 17.3 shows examples of the relationship between the SPBR register and BRDV1 and BRDV0 bit settings.

Table 17.3 Relationship between SPBR and BRDV1 and BRDV0 Settings

SPBR (n)	BRDV[1:0] (N)	Division ratio	Bit Rate			
			Pϕ=16MHz	Pϕ=20MHz	Pϕ=32MHz	Pϕ=40MHz
0	0	2	8.0 Mbps	10.0 Mbps	—	—
1	0	4	4.0 Mbps	5.0 Mbps	8.0 Mbps	10.0 Mbps
2	0	6	2.67 Mbps	3.3 Mbps	5.33 Mbps	6.67 Mbps
3	0	8	2.0 Mbps	2.5 Mbps	4.0 Mbps	5.0 Mbps
4	0	10	1.6 Mbps	2.0 Mbps	3.2 Mbps	4.0 Mbps
5	0	12	1.33 Mbps	1.67 Mbps	2.67 Mbps	3.33 Mbps
5	1	24	667 kbps	833 kbps	1.33 Mbps	1.67 Mbps
5	2	48	333 kbps	417 kbps	667 kbps	8.33 kbps
5	3	96	167 kbps	208 kbps	333 kbps	417 kbps
255	3	4096	3.9 kbps	4.9 kbps	7.8 kbps	9.8 kbps

Note: — setting prohibited

17.3.9 RSPI Clock Delay Register (SPCKD)

SPCKD sets a period from the beginning of SSL signal assertion to RSPCK oscillation (RSPCK delay) when the SCKDEN bit in the RSPI command register (SPCMD) is 1. SPCKD can be read from or written to by the CPU. If the contents of SPCKD are changed by the CPU while the MSTR and SPE bits in the RSPI control register (SPCR) are 1 with the RSPI function in master mode enabled, operation cannot be guaranteed.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	SCKDL[2:0]		
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved The write value should always be 0. Otherwise, operation cannot be guaranteed.
2 to 0	SCKDL[2:0]	000	R/W	RSPCK Delay Setting These bits set an RSPCK delay value when the SCKDEN bit in SPCMD is 1. 000: 1 RSPCK 001: 2 RSPCK 010: 3 RSPCK 011: 4 RSPCK 100: 5 RSPCK 101: 6 RSPCK 110: 7 RSPCK 111: 8 RSPCK

17.3.10 SPI Slave Select Negation Delay Register (SSLND)

SSLND sets a period (SSL negation delay) from the transmission of a final RSPCK edge to the negation of the SSL signal during a serial transfer by the RSPI in master mode. SSLND can be read from or written to by the CPU. If the contents of SSLND are changed by the CPU while the MSTR and SPE bits in the RSPI control register (SPCR) are 1 with the RSPI function in master mode enabled, operation cannot be guaranteed.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	SLNDL[2:0]		
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved The write value should always be 0. Otherwise, operation cannot be guaranteed.
2 to 0	SLNDL[2:0]	000	R/W	SSL Negation Delay Setting These bits set an SSL negation delay value when the RSPI is in master mode. 000: 1 RSPCK 001: 2 RSPCK 010: 3 RSPCK 011: 4 RSPCK 100: 5 RSPCK 101: 6 RSPCK 110: 7 RSPCK 111: 8 RSPCK

17.3.11 RSPI Next-Access Delay Register (SPND)

SPND sets a non-active period (next-access delay) after termination of a serial transfer when the SPNDEN bit in the RSPI command register (SPCMD) is 1. SPND can be read from or written to by the CPU. If the contents of SPND are changed by the CPU while the MSTR and SPE bits in the RSPI control register (SPCR) are 1 with the RSPI function in master mode enabled, operation cannot be guaranteed.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	SPNDL[2:0]		
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved The write value should always be 0. Otherwise, operation cannot be guaranteed.
2 to 0	SPNDL[2:0]	000	R/W	RSPI Next-Access Delay Setting These bits set a next-access delay when the SPNDEN bit in SPCMD is 1. 000: 1 RSPCK + 2 Pϕ 001: 2 RSPCK + 2 Pϕ 010: 3 RSPCK + 2 Pϕ 011: 4 RSPCK + 2 Pϕ 100: 5 RSPCK + 2 Pϕ 101: 6 RSPCK + 2 Pϕ 110: 7 RSPCK + 2 Pϕ 111: 8 RSPCK + 2 Pϕ

17.3.12 RSPI Command Register (SPCMD)

Each channel has eight RSPI command registers (SPCMD0 to SPCMD7). SPCMD0 to SPCMD7 are used to set a transfer format for the RSPI in master mode. Some of the bits in SPCMD0 are used to set a transfer mode for the RSPI in slave mode. The RSPI in master mode sequentially references SPCMD0 to SPCMD7 according to the settings in bits SPSLN2 to SPSLN0 in the RSPI sequence control register (SPSCR), and executes the serial transfer that is set in the referenced SPCMD.

SPCMD can be read from or written to by the CPU. If the CPU rewrites the SPCMD that is referenced by the RSPI while the RSPI is performing serial transfer in master mode, or during SSL negation delay or next-access delay, correct operation cannot be guaranteed. SPCMD that is referenced by the RSPI in master mode can be checked by means of bits SPCP2 to SPCP0 in the RSPI sequence status register (SPSSR). When the RSPI function in slave mode is enabled, operation cannot be guaranteed if the value set in SPCMD0 is changed by the CPU.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SCK DEN	SLN DEN	SPN DEN	LSBF	SPB[3:0]				SSLKP	SSLA[2:0]			BRDV[1:0]		CPOL	CPHA
Initial value:	0	0	0	0	0	1	1	1	0	0	0	0	1	1	0	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	SCKDEN	0	R/W	RSPCK Delay Setting Enable Sets the period from the time the RSPI in master mode sets the SSL signal active until the RSPI oscillates RSPCK (RSPCK delay). If the SCKDEN bit is 0, the RSPI sets the RSPCK delay to 1 RSPCK. If the SCKDEN bit is 1, the RSPI starts the oscillation of RSPCK at an RSPCK delay in compliance with RSPCK delay register (SPCKD) settings. To use the RSPI in slave mode, the SCKDEN bit should be set to 0. 0: An RSPCK delay of 1 RSPCK 1: An RSPCK delay equal to SPCKD settings.

Bit	Bit Name	Initial Value	R/W	Description
14	SLNDEN	0	R/W	SSL Negation Delay Setting Enable Sets the period (SSL negation delay) from the time the master mode RSPI stops RSPCK oscillation until the RSPI sets the SSL signal inactive. If the SLNDEN bit is 0, the RSPI sets the SSL negation delay to 1 RSPCK. If the SLNDEN bit is 1, the RSPI negates the SSL signal at an SSL negation delay in compliance with slave select negation delay register (SSLND) settings. To use the RSPI in slave mode, the SLNDEN bit should be set to 0. 0: An SSL negation delay of 1 RSPCK 1: An SSL negation delay equal to SSLND settings.
13	SPNDEN	0	R/W	RSPI Next-Access Delay Enable Sets the period from the time the RSPI in master mode terminates a serial transfer and sets the SSL signal inactive until the RSPI enables the SSL signal assertion for the next access (next-access delay). If the SPNDEN bit is 0, the RSPI sets the next-access delay to 1 RSPCK + 2Pϕ. If the SPNDEN bit is 1, the RSPI inserts a next-access delay in compliance with RSPI next-access delay register (SPND) settings. To use the RSPI in slave mode, the SPNDEN bit should be set to 0. 0: A next-access delay of 1 RSPCK + 2 Pϕ 1: A next-access delay equal to SPND settings.
12	LSBF	0	R/W	RSPI LSB First Sets the data format of the RSPI in master mode or slave mode to MSB first or LSB first. 0: MSB first 1: LSB first

Bit	Bit Name	Initial Value	R/W	Description
11 to 8	SPB[3:0]	0111	R/W	SRPI Data Length Setting These bits set a transfer data length for the RSPI in master mode or slave mode. 0000 to 0111: 8 bits 1000: 9 bits 1001: 10 bits 1010: 11 bits 1011: 12 bits 1100: 13 bits 1101: 14 bits 1110: 15 bits 1111: 16 bits
7	SSLKP	0	R/W	SSL Signal Level Keeping When the RSPI in master mode performs a serial transfer, this bit specifies whether the SSL signal level for the current command is to be kept or negated between the SSL negation timing associated with the current command and the SSL assertion timing associated with the next command. To use the RSPI in slave mode, the SSLKP bit should be set to 0. 0: Negates all SSL signals upon completion of transfer. 1: Keeps the SSL signal level from the end of the transfer until the beginning of the next access.

Bit	Bit Name	Initial Value	R/W	Description
6 to 4	SSLA[2:0]	000	R/W	SSL Signal Assertion Setting These bits control the SSL signal assertion when the RSPI performs serial transfers in master mode. Setting the SSLA[2:0] controls the assertion for the signals SSL7 to SSL0. When an SSL signal is asserted, its polarity is determined by the set value in the corresponding SSLP (RSPI slave select polarity register). When SSLA[2:0] are set to B'000 in multi-master mode, serial transfers are performed with all the SLL signals in the neagted state (as SSL0 acts as input). Also when setting the values B'100 to B'111 to the bits SSLA[2:0] of the registers SPCMDB0 to SPCMDB7 and SPCMDC0 to SPCMDC7, serial transfers are performed with the SSL signals in the negated state (as channels B and C lack SSL4 to SSL7). When using the RSPI in slave mode, set B'000 to SSLA[2:0]. 000: SSL0 001: SSL1 010: SSL2 011: SSL3 100: SSL4 101: SSL5 110: SSL6 111: SSL7

Bit	Bit Name	Initial Value	R/W	Description
3, 2	BRDV[1:0]	11	R/W	Bit Rate Division Setting These bits are used to determine the bit rate. A bit rate is determined by combinations of bits BRDV1 and BRDV 0 and the settings in the RSPI bit rate register (SPBR) (see section 17.3.8, RSPI Bit Rate Register (SPBR)). The settings in SPBR determine the base bit rate. The settings in bits BRDV1 and BRDV0 are used to select a bit rate which is obtained by dividing the base bit rate by 1, 2, 4, or 8. In the bits SPCMD0 to SPCMD7 different BRDV1 and BRDV0 settings can be specified. This permits the execution of serial transfers at a different bit rate for each command. 00: Select the base bit rate 01: Select the base bit rate divided by 2 10: Select the base bit rate divided by 4 11: Select the base bit rate divided by 8
1	CPOL	0	R/W	RSPCK Polarity Setting Sets an RSPCK polarity of the RSPI in master or slave mode. Data communications between RSPI modules require the same RSPCK polarity setting between the modules. 0: RSPCK = 0 when idle 1: RSPCK = 1 when idle
0	CPHA	1	R/W	RSPCK Phase Setting Sets an RSPCK phase of the RSPI in master or slave mode. Data communications between RSPI modules require the same RSPCK phase setting between the modules. 0: Data sampling on odd edge, data variation on even edge 1: Data variation on odd edge, data sampling on even edge

17.4 Operation

In this section, the serial transfer period means a period from the beginning of driving valid data to the fetching of the final valid data.

17.4.1 Overview of RSPI Operations

The RSPI is capable of synchronous serial transfers in slave, single-master, and multi-master modes. A particular mode of the RSPI can be selected by using the MSTR and MODFEN bits in the RSPI control register (SPCR). Table 17.4 gives the relationship between RSPI modes and SPCR settings, and a description of each mode.

Table 17.4 Relationship between RSPI Modes and SPCR and Description of Each Mode

Item		Slave Mode	Single-Master Mode	Multi-Master Mode
MSTR bit setting		0	1	1
MODFEN bit setting		0, 1	0	1
RSPCK signal		Input	Output	Output/Hi-Z
MOSI signal		Input	Output	Output/Hi-Z
MISO signal		Output/Hi-Z	Input	Input
SSL0 signal		Input	Output	Input
SSL1 to SSL7 signals		Hi-Z	Output	Output/Hi-Z
Output pin mode		CMOS/open-drain	CMOS/open-drain	CMOS/open-drain
SSL polarity modification function		Supported	Supported	Supported
Transfer rate	One-time multiplied peripheral clock	Up to 2.5 MHz	Up to 10.0 MHz	Up to 10.0 MHz
	Two-times multiplied peripheral clock	Up to 2.5 MHz	Up to 10.0 MHz	Up to 10.0 MHz
Clock source		RSPCK input	On-chip baud rate generator	On-chip baud rate generator
Clock polarity		Two	Two	Two
Clock phase		Two	Two	Two
First transfer bit		MSB/LSB	MSB/LSB	MSB/LSB

Item	Slave Mode	Single-Master Mode	Multi-Master Mode
Transfer data length	8 to 16 bits	8 to 16 bits	8 to 16 bits
Burst transfer	Possible (CPHA = 1)	Possible (CPHA = 0, 1)	Possible (CPHA = 0, 1)
RSPCK delay control	Not supported	Supported	Supported
SSL negation delay control	Not supported	Supported	Supported
Next-access delay control	Not supported	Supported	Supported
Transfer starting method	SSL input active or RSPCK oscillation	Started by the A-DMAC: write to SPDR Started by the CPU: clear SPTEF	Started by the A-DMAC: write to SPDR Started by the CPU: clear SPTEF
Sequence control	Not supported	Supported	Supported
Transmission buffer empty detection	Supported	Supported	Supported
Receive buffer full detection	Supported	Supported	Supported
Overrun error detection	Supported	Supported	Supported
Mode fault error detection	Supported (MODFEN = 1)	Not supported	Supported

17.4.2 Controlling RSPI Pins

According to the MSTR and MODFEN bits in the RSPI control register (SPCR) and the SPOM bit in the RSPI pin control register (SPPCR), the RSPI can automatically switch pin directions and output modes. Table 17.5 shows the relationship between pin states and bit settings.

Table 17.5 Relationship between Pin States and Bit Settings

		Pin State*[1]	
Mode	**Pin**	**SPOM = 0**	**SPOM = 1**
Single-master mode (MSTR = 1, MODFEN = 0)	RSPCK	CMOS output	Open-drain output
	SSL0 to SSL7	CMOS output	Open-drain output
	MOSI	CMOS output	Open-drain output
	MISO	Input	Input
Multi-master mode (MSTR = 1, MODFEN = 1)	RSPCK*[2]	CMOS output/Hi-Z	Open-drain output/Hi-Z
	SSL0	Input	Input
	SSL1 to SSL7	CMOS output/Hi-Z	Open-drain output/Hi-Z
	MOSI*[2]	CMOS output/Hi-Z	Open-drain output/Hi-Z
	MISO	Input	Input
Slave mode (MSTR = 0)	RSPCK	Input	Input
	SSL0	Input	Input
	SSL1 to SSL7	Hi-Z	Hi-Z
	MOSI	Input	Input
	MISO*[3]	CMOS output/Hi-Z	Open-drain output/Hi-Z

Notes: 1. RSPI settings are not indicated in the multiplex pins for which the RSPI function is not selected.

2. When SSL0 is at the active level, the pin state is Hi-Z.

3. When SSL0 is at the non-active level or the SPE bit in SPCR is 0, the pin state is Hi-Z.

The RSPI in master mode determines MOSI signal values during the SSL negation period (including the SSL retention period during a burst transfer) according to MOIFE and MOIFV bit settings in SPPCR, as shown in table 17.6.

Table 17.6 MOSI Signal Value Determination during SSL Negation Period

MOIFE	MOIFV	MOSI Signal Value during SSL Negation Period
0	0, 1	Final data from previous transfer
1	0	Always 0
1	1	Always 1

17.4.3 RSPI System Configuration Example

(1) Single Master/Single Slave (with this LSI Acting as Master)

Figure 17.2 shows a single-master/single-slave RSPI system configuration example when this LSI is used as a master. In the single-master/single-slave configuration, the SSL0 to SSL7 output of this LSI (master) is not used. The SSL input of the RSPI slave is fixed to the low level, and the RSPI slave is always maintained in a select state. In the transfer format corresponding to the case where the CPHA bit in the RSPI control register (SPCR) is 0, there are slave devices for which the SSL signal cannot be fixed to the active level. In situations where the SSL signal cannot be fixed, the SSL output of this LSI should be connected to the SSL input of the slave device.

This LSI (master) always drives the RSPCK and MOSI. The RSPI slave always drives the MISO.

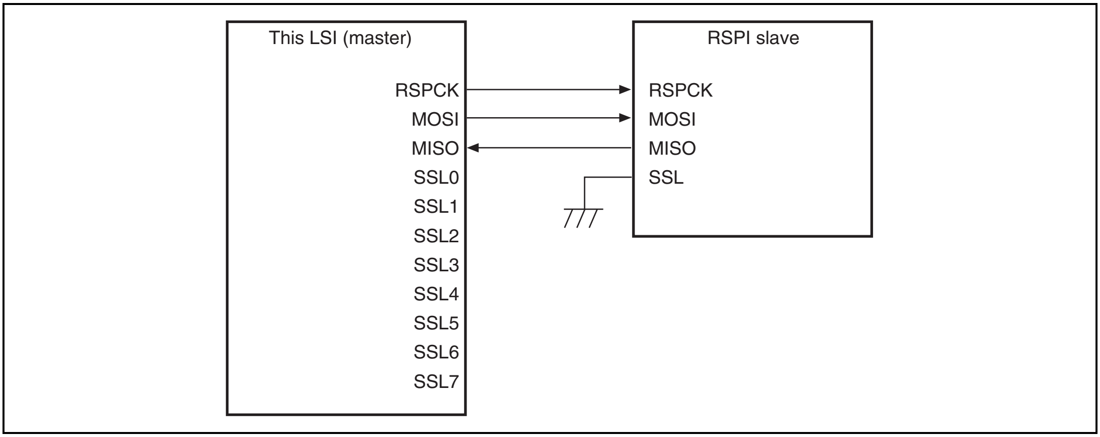


Figure 17.2 Single-Master/Single-Slave Configuration Example (This LSI = Master)

(2) Single Master/Single Slave (with this LSI Acting as Slave)

Figure 17.3 shows a single-master/single-slave RSPI system configuration example when this LSI is used as a slave. When this LSI is to operate as a slave, the SSL0 pin is used as SSL input. The RSPI master always drives the RSPCK and MOSI. This LSI (slave) always drives the MISO.

In the single-slave configuration in which the CPHA bit in the RSPI command register (SPCMD) is set to 1, the SSL0 input of this LSI (slave) is fixed to the low level, this LSI (slave) is always maintained in a selected state, and in this manner it is possible to execute serial transfer (figure 17.4).

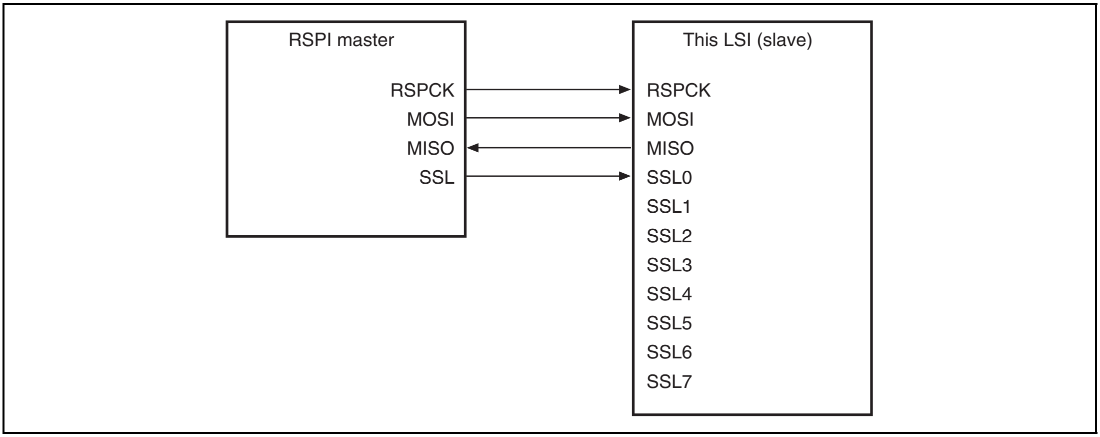


Figure 17.3 Single-Master/Single-Slave Configuration Example (This LSI = Slave)

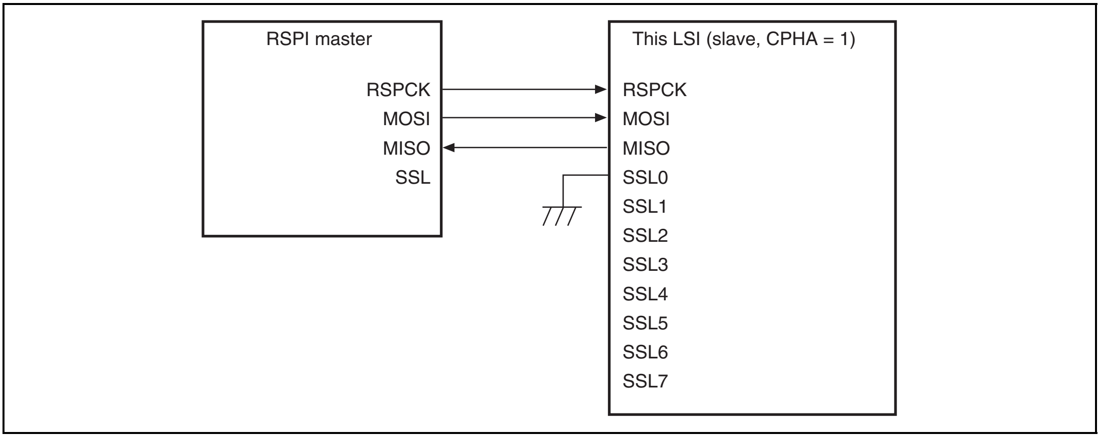


Figure 17.4 Single-Master/Single-Slave Configuration Example (This LSI = Slave, CPHA = 1)

(3) Single Master/Multi-Slave (with this LSI Acting as Master)

Figure 17.5 shows a single-master/multi-slave RSPI system configuration example when this LSI is used as a master. In the example of figure 17.5, the RSPI system is comprised of an LSI (master) and four slaves (RSPI slave 0 to RSPI slave 3).

The RSPCK and MOSI outputs of this LSI (master) are connected to the RSPCK and MOSI inputs of RSPI slave 0 to RSPI slave 3. The MISO outputs of RSPI slave 0 to RSPI slave 3 are all connected to the MISO input of this LSI (master). SSL0 to SSL3 outputs of this LSI (master) are connected to the SSL inputs of RSPI slave 0 to RSPI slave 3, respectively. In this configuration example, because there are four RSPI slaves, the SSL4 to SSL7 outputs of this LSI (master) are not used.

This LSI (master) always drives RSPCK, MOSI, and SSL0 to SSL3. Of the RSPI slave 0 to RSPI slave 3, the slave that receives low level input into the SSL input drives MISO.

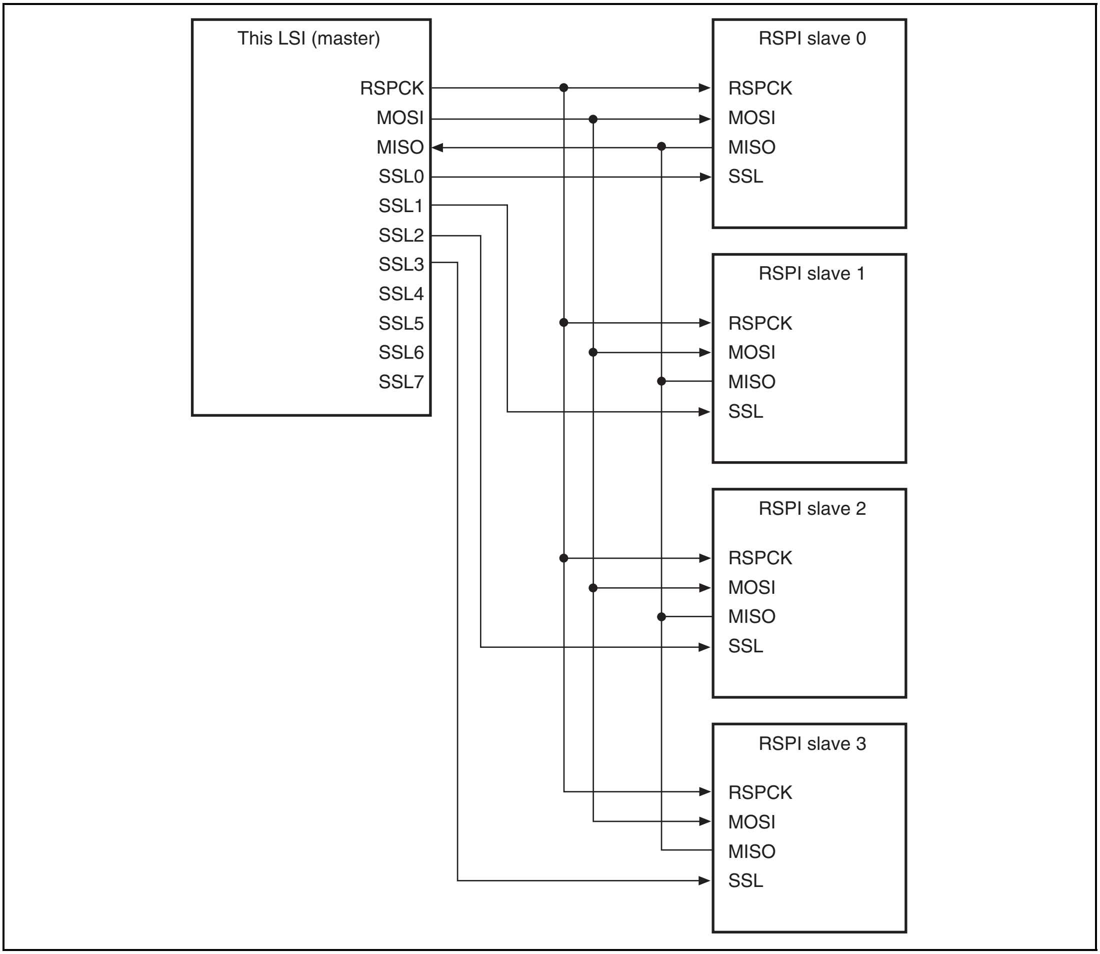


Figure 17.5 Single-Master/Multi-Slave Configuration Example (This LSI = Master)

(4) Single Master/Multi-Slave (with this LSI Acting as Slave)

Figure 17.6 shows a single-master/multi-slave RSPI system configuration example when this LSI is used as a slave. In the example of figure 17.6, the RSPI system is comprised of an RSPI master and the two LSIs (slave X and slave Y).

The RSPCK and MOSI outputs of the RSPI master are connected to the RSPCK and MOSI inputs of these LSIs (slave X and slave Y). The MISO outputs of these LSIs (slave X and slave Y) are all connected to the MISO input of the RSPI master. SSLX and SSLY outputs of the RSPI master are connected to the SSL0 inputs of these LSIs (slave X and slave Y), respectively.

The RSPI master always drives RSPCK, MOSI, SSLX, and SSLY. Of this LSIs (slave X and slave Y), the slave that receives low level input into the SSL0 input drives MISO.

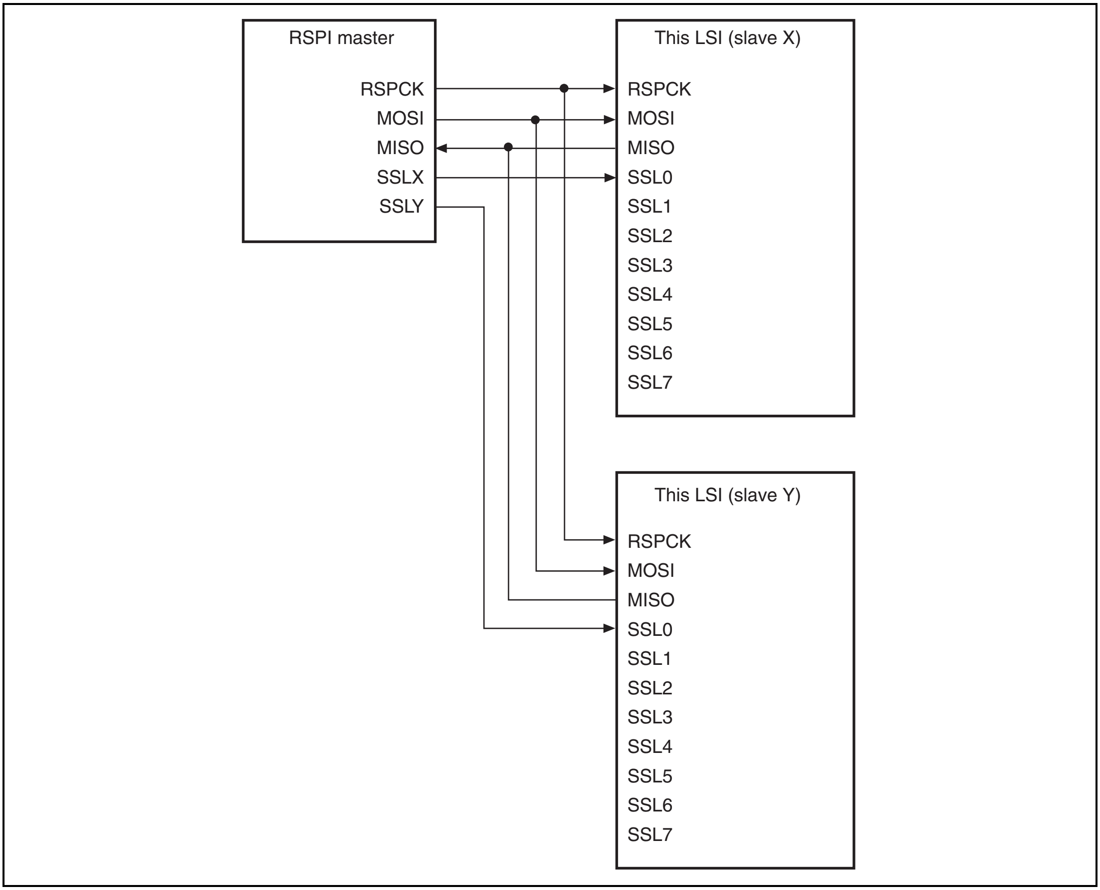


Figure 17.6 Single-Master/Multi-Slave Configuration Example (This LSI = Slave)

(5) Multi-Master/Multi-Slave (with this LSI Acting as Master)

Figure 17.7 shows a multi-master/multi-slave RSPI system configuration example when this LSI is used as a master. In the example of figure 17.7, the RSPI system is comprised of the two LSIs (master X, master Y) and two RSPI slaves (RSPI slave 1, RSPI slave 2).

The RSPCK and MOSI outputs of this LSI (master X, master Y) are connected to the RSPCK and MOSI inputs of RSPI slaves 1 and 2. The MISO outputs of RSPI slaves 1 and 2 are connected to the MISO inputs of this LSI (master X, master Y). Any generic port Y output from this LSI (master X) is connected to the SSL0 input of this LSI (master Y). Any generic port X output of this LSI (master Y) is connected to the SSL0 input of this LSI (master X). The SSL1 and SSL2 outputs of this LSI (master X, master Y) are connected to the SSL inputs of the RSPI slaves 1 and 2. In this configuration example, because the system can be comprised solely of SSL0 input, and SSL1 and SSL2 outputs for slave connections, the outputs SSL3 to SSL7 of this LSI are not required.

This LSI drives RSPCK, MOSI, SSL1, and SSL2 when the SSL0 input level is 1. When the SSL0 input level is 0, this LSI detects a mode fault error, sets RSPCK, MOSI, SSL1, and SSL2 to Hi-Z, and releases the RSPI bus right to the other master. Of the RSPI slaves 1 and 2, the slave that receives a level-0 input into the SSL input drives MISO.

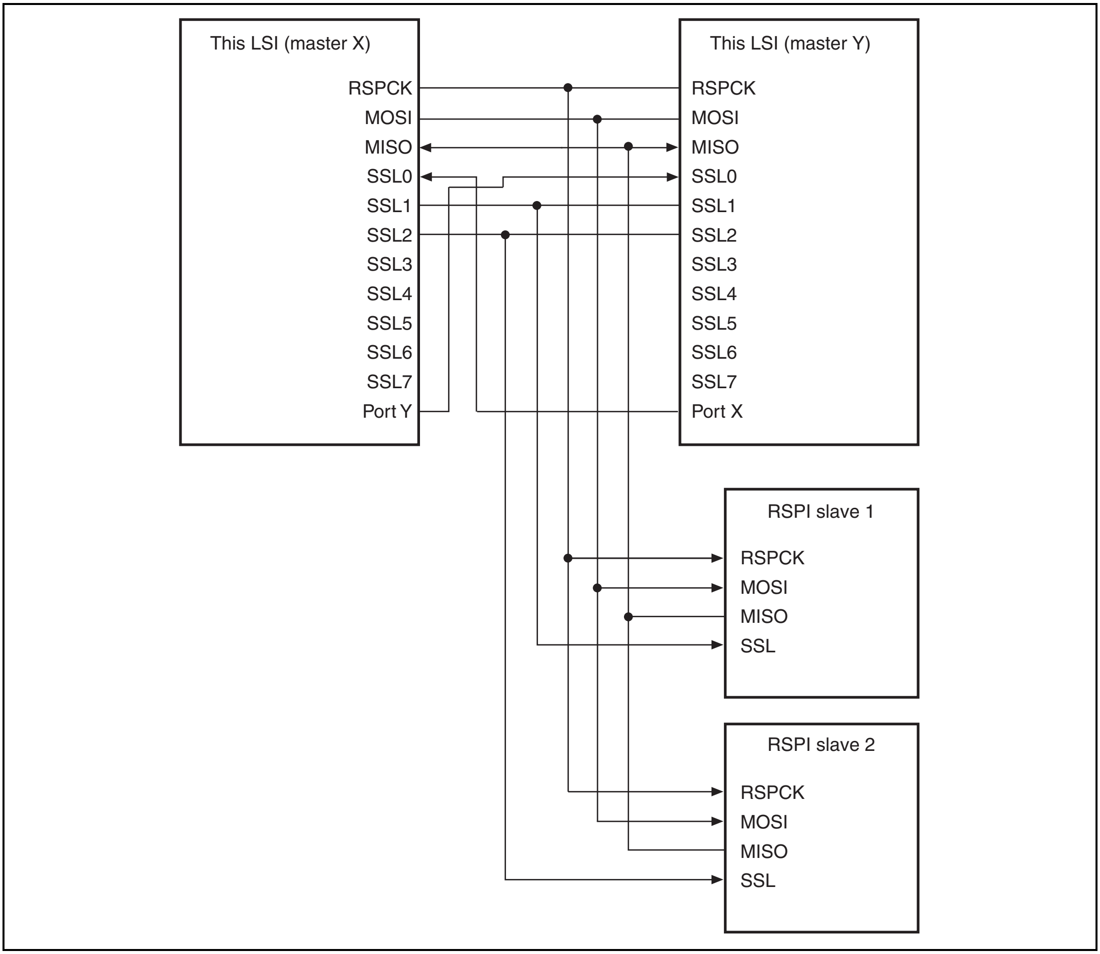


Figure 17.7 Multi-Master/Multi-Slave Configuration Example (This LSI = Master)

17.4.4 Transfer Format

(1) CPHA = 0

Figure 17.8 shows an example transfer format for the serial transfer of 8-bit data when the CPHA bit in the RSPI command register (SPCMD) is 0. In Figure 17.8, RSPCK (CPOL = 0) indicates the RSPCK signal waveform when the CPOL bit in SPCMD is 0; RSPCK (CPOL = 1) indicates the RSPCK signal waveform when the CPOL bit is 1. The sampling timing represents the timing at which the RSPI fetches serial transfer data into the shift register. The input/output directions of the signals depend on the RSPI settings. For details, see section 17.4.2, Controlling RSPI Pins.

When the CPHA bit is 0, the driving of valid data to the MOSI and MISO signals commences at an SSL signal assertion timing. The first RSPCK signal change timing that occurs after the SSL signal assertion becomes the first transfer data fetching timing. After this timing, data is sampled at every 1 RSPCK cycle. The change timing for MOSI and MISO signals is always 1/2 RSPCK cycle after the transfer data fetch timing. The settings in the CPOL bit do not affect the RSPCK signal operation timing; they only affect the signal polarity.

t1 denotes a period from an SSL signal assertion to RSPCK oscillation (RSPCK delay). t2 denotes a period from the cessation of RSPCK oscillation to an SSL signal negation (SSL negation delay). t3 denotes a period in which SSL signal assertion is suppressed for the next transfer after the end of serial transfer (next-access delay). t1, t2, and t3 are controlled by a master device running on the RSPI system. For a description of t1, t2, and t3 when the RSPI of this LSI is in master mode, see section 17.4.9, Master Mode Operation.

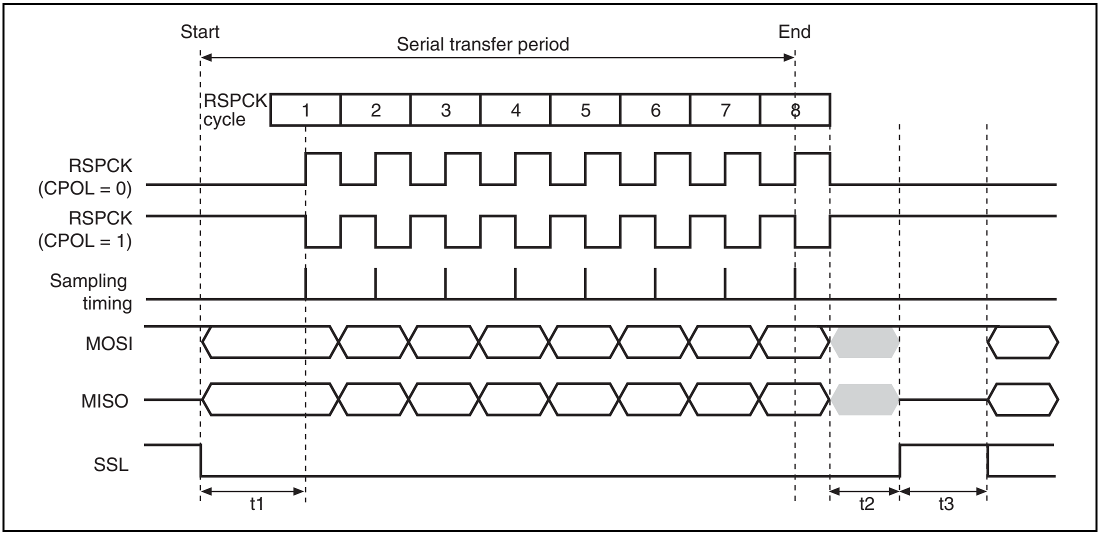


Figure 17.8 RSPI Transfer Format (CPHA = 0)

(2) CPHA = 1

Figure 17.9 shows an example transfer format for the serial transfer of 8-bit data when the CPHA bit in the RSPI command register (SPCMD) is 1. In figure 17.9, RSPCK (CPOL = 0) indicates the RSPCK signal waveform when the CPOL bit in SPCMD is 0; RSPCK (CPOL = 1) indicates the RSPCK signal waveform when the CPOL bit is 1. The sampling timing represents the timing at which the RSPI fetches serial transfer data into the shift register. The input/output directions of the signals depend on the RSPI modes (master or slave). For details, see section 17.4.2, Controlling RSPI Pins.

When the CPHA bit is 1, the driving of invalid data to the MOSI and MISO signals commences at an SSL signal assertion timing. The driving of valid data to the MOSI and MISO signals commences at the first RSPCK signal change timing that occurs after the SSL signal assertion. After this timing, data is updated at every 1 RSPCK cycle. The transfer data fetch timing is always 1/2 RSPCK cycle after the data update timing. The settings in the CPOL bit do not affect the RSPCK signal operation timing; they only affect the signal polarity.

t1, t2, and t3 are the same as those in the case of CPHA = 0. For a description of t1, t2, and t3 when the RSPI of this LSI is in master mode, see section 17.4.9, Master Mode Operation.

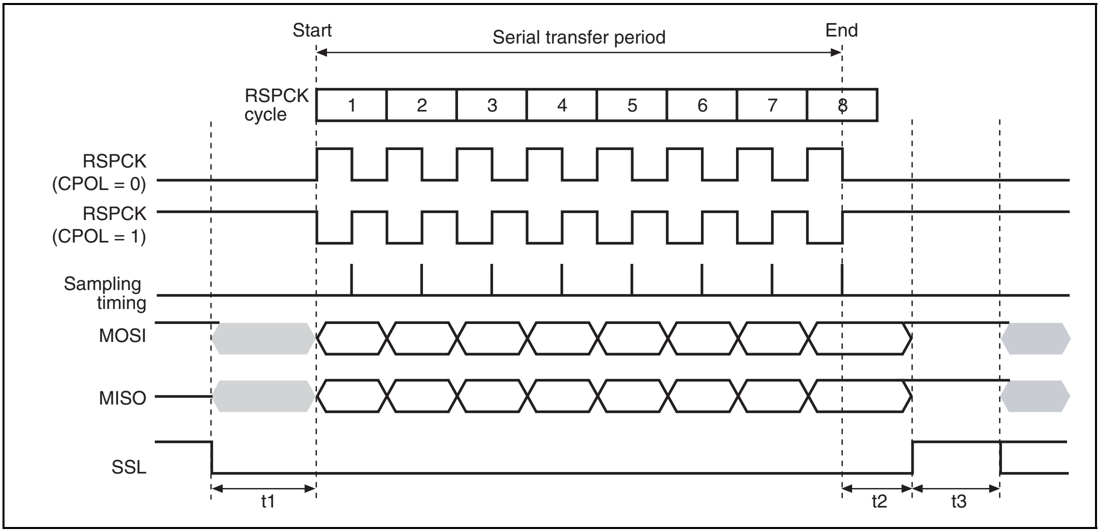


Figure 17.9 RSPI Transfer Format (CPHA = 1)

17.4.5 Data Format

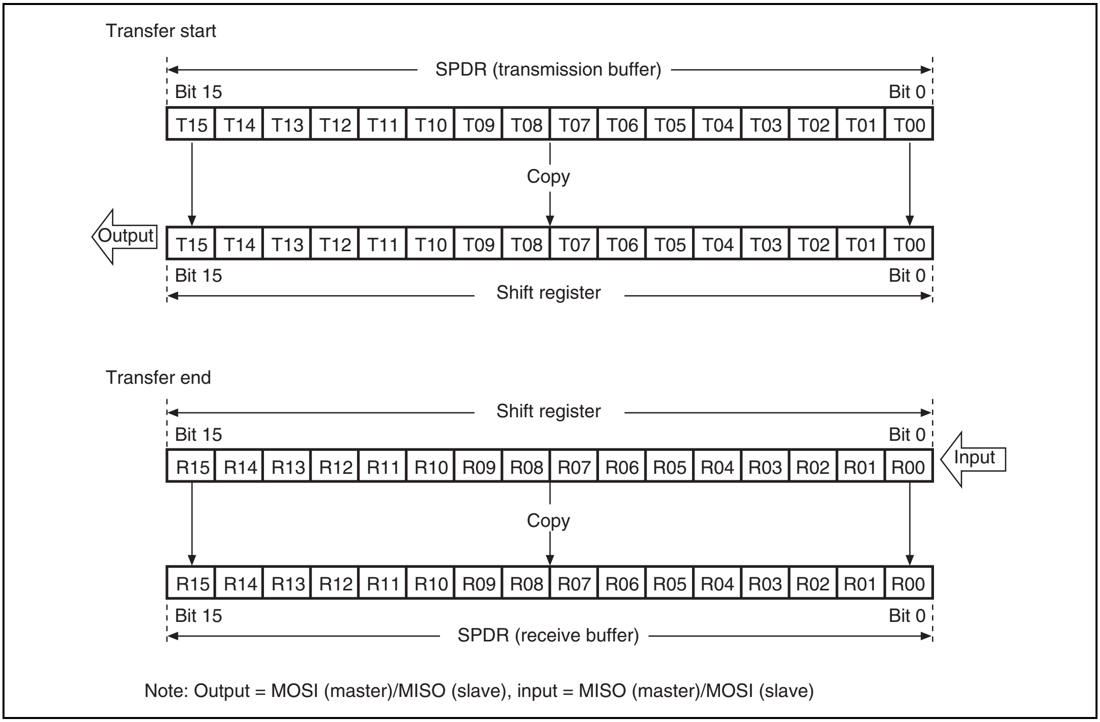
The RSPI's data format depends on the settings in the RSPI command register (SPCMD). Irrespective of MSB/LSB first, the RSPI treats the range from the LSB of the RSPI data register (SPDR) to the assigned data length as transfer data.

(1) MSB First Transfer (16-Bit Data)

Figure 17.10 shows the operation of the RSPI data register (SPDR) and the shift register when the RSPI performs a 16-bit data length MSB-first data transfer.

The CPU or the A-DMAC writes T15 to T00 to the transmission buffer of SPDR. If the SPTEF bit in the RSPI status register (SPSR) is 0 and the shift register is empty, the RSPI copies the data in the transmission buffer of SPDR to the shift register, and fully populates the shift register. When serial transfer starts, the RSPI outputs data from the MSB (bit 15) of the shift register, and shifts in the data from the LSB (bit 0) of the shift register. When the RSPCK cycle required for the serial transfer of 16 bits has passed, data R15 to R00 is stored in the shift register. In this state, the RSPI copies the data from the shift register to the receive buffer of SPDR, and empties the shift register.

If another serial transfer is started before the CPU or the A-DMAC writes to the transmission buffer of SPDR, received data R15 to R00 is shifted out from the shift register.


Transfer start
SPDR (transmission buffer)
Bit 15
Bit 0
T15 T14 T13 T12 T11 T10 T09 T08 T07 T06 T05 T04 T03 T02 T01 T00
Copy
Output
T15 T14 T13 T12 T11 T10 T09 T08 T07 T06 T05 T04 T03 T02 T01 T00
Bit 15
Bit 0
Shift register
Transfer end
Shift register
Bit 15
Bit 0
R15 R14 R13 R12 R11 R10 R09 R08 R07 R06 R05 R04 R03 R02 R01 R00
Input
Copy
R15 R14 R13 R12 R11 R10 R09 R08 R07 R06 R05 R04 R03 R02 R01 R00
Bit 15
Bit 0
SPDR (receive buffer)

Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

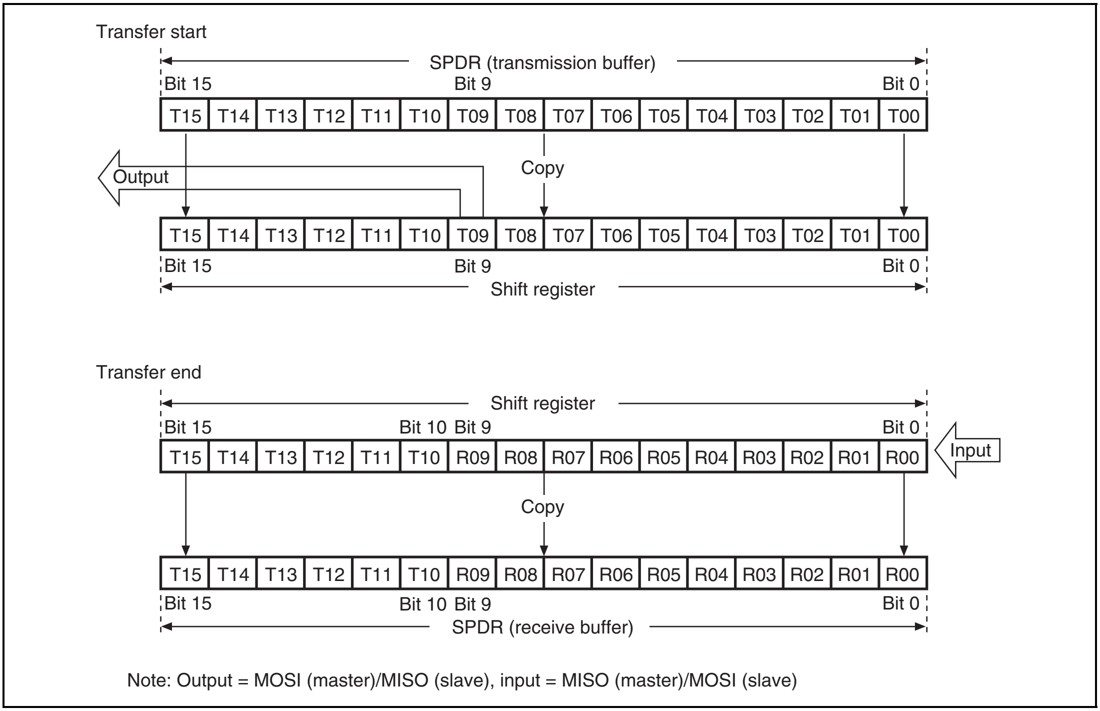
Figure 17.10 MSB First Transfer (16-Bit Data)

(2) MSB First Transfer (10-Bit Data)

Figure 17.11 shows the operation of the RSPI data register (SPDR) and the shift register when the RSPI performs a 10-bit data length MSB-first data transfer.

The CPU or the A-DMAC writes T15 to T00 to the transmission buffer of SPDR. If the SPTEF bit in the RSPI status register (SPSR) is 0 and the shift register is empty, the RSPI copies the data in the transmission buffer of SPDR to the shift register, and fully populates the shift register. When serial transfer starts, the RSPI outputs data from bit 9 of the shift register, and shifts in the data from the LSB (bit 0) of the shift register. When the RSPCK cycle required for the serial transfer of 10 bits has passed, received data R09 to R00 is stored in bits 9 to 0 of the shift register. After completion of the serial transfer, data that existed before the transfer is retained in bits 15 to 10 in the shift register. In this state, the RSPI copies the data from the shift register to the receive buffer of SPDR, and empties the shift register.

If another serial transfer is started before the CPU or the A-DMAC writes to the transmission buffer of SPDR, received data R09 to R00 is shifted out from the shift register.


Transfer start
SPDR (transmission buffer)
Bit 15
Bit 9
Bit 0
T15 T14 T13 T12 T11 T10 T09 T08 T07 T06 T05 T04 T03 T02 T01 T00
Output
Copy
T15 T14 T13 T12 T11 T10 T09 T08 T07 T06 T05 T04 T03 T02 T01 T00
Bit 15
Bit 9
Bit 0
Shift register
Transfer end
Shift register
Bit 15
Bit 10
Bit 9
Bit 0
T15 T14 T13 T12 T11 T10 R09 R08 R07 R06 R05 R04 R03 R02 R01 R00
Input
Copy
T15 T14 T13 T12 T11 T10 R09 R08 R07 R06 R05 R04 R03 R02 R01 R00
Bit 15
Bit 10
Bit 9
Bit 0
SPDR (receive buffer)

Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

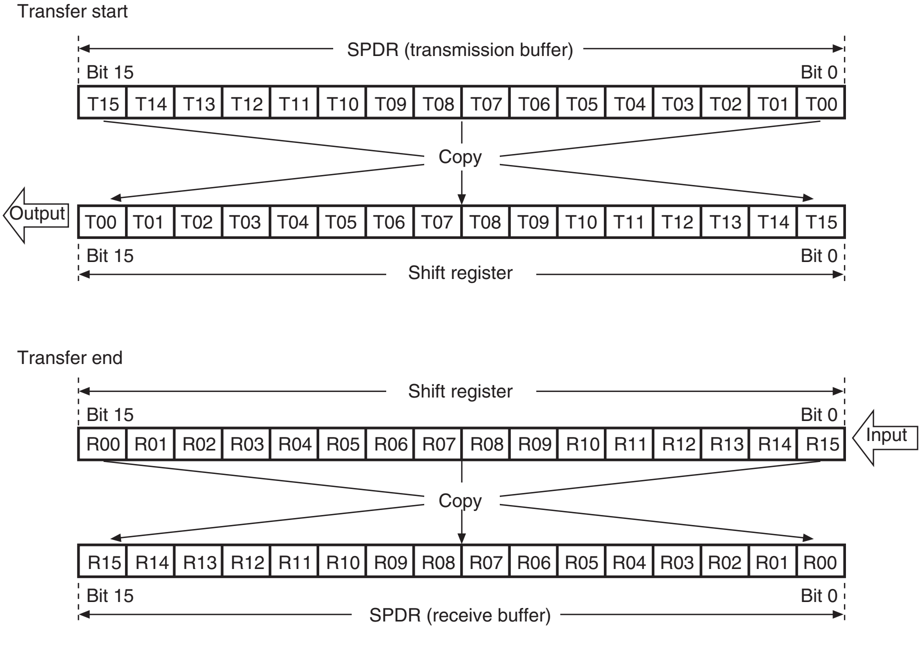
Figure 17.11 MSB First Transfer (10-Bit Data)

(3) LSB First Transfer (16-Bit Data)

Figure 17.12 shows the operation of the RSPI data register (SPDR) and the shift register when the RSPI performs a 16-bit data length LSB-first data transfer.

The CPU or the A-DMAC writes T15 to T00 to the transmission buffer of SPDR. If the SPTEF bit in the RSPI status register (SPSR) is 0 and the shift register is empty, the RSPI reverses the order of the bits of the data in the transmission buffer of SPDR, copies it to the shift register, and fully populates the shift register. When serial transfer starts, the RSPI outputs data from the MSB (bit 15) of the shift register, and shifts in the data from the LSB (bit 0) of the shift register. When the RSPCK cycle required for the serial transfer of 16 bits has passed, data R00 to R15 is stored in the shift register. In this state, the RSPI copies the data, in which the order of the bits is reversed, from the shift register to the receive buffer of SPDR, and empties the shift register.

If another serial transfer is started before the CPU or the A-DMAC writes to the transmission buffer of SPDR, received data R00 to R15 is shifted out from the shift register.


Transfer start
SPDR (transmission buffer)
Bit 15
Bit 0
T15 T14 T13 T12 T11 T10 T09 T08 T07 T06 T05 T04 T03 T02 T01 T00
Copy
Output
T00 T01 T02 T03 T04 T05 T06 T07 T08 T09 T10 T11 T12 T13 T14 T15
Bit 15
Bit 0
Shift register
Transfer end
Shift register
Bit 15
Bit 0
R00 R01 R02 R03 R04 R05 R06 R07 R08 R09 R10 R11 R12 R13 R14 R15
Input
Copy
R15 R14 R13 R12 R11 R10 R09 R08 R07 R06 R05 R04 R03 R02 R01 R00
Bit 15
Bit 0
SPDR (receive buffer)

Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

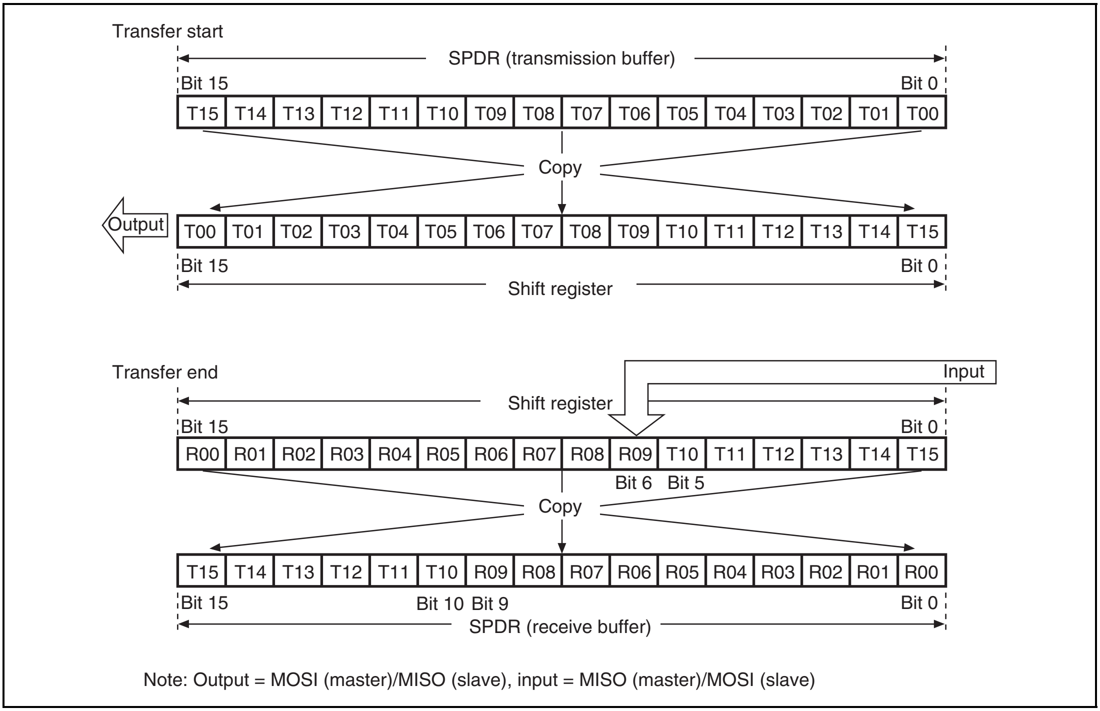
Figure 17.12 LSB First Transfer (16-Bit Data)

(4) LSB First Transfer (10-Bit Data)

Figure 17.13 shows the operation of the RSPI data register (SPDR) and the shift register when the RSPI performs a 10-bit data length LSB-first data transfer.

The CPU or the A-DMAC writes T15 to T00 to the transmission buffer of SPDR. If the SPTEF bit in the RSPI status register (SPSR) is 0 and the shift register is empty, the RSPI reverses the order of the bits of the data in the transmission buffer of SPDR, copies it to the shift register, and fully populates the shift register. When serial transfer starts, the RSPI outputs data from the MSB (bit 15) of the shift register, and shifts in the data from bit 6 of the shift register. When the RSPCK cycle required for the serial transfer of 10 bits has passed, received data R00 to R09 is stored in bits 15 to 6 of the shift register. After completion of the serial transfer, data that existed before the transfer is retained in bits 5 to 0 of the shift register. In this state, the RSPI copies the data, in which the order of the bits is reversed, from the shift register to the receive buffer of SPDR, and empties the shift register.

If another serial transfer is started before the CPU or the A-DMAC writes to the transmission buffer of SPDR, received data R00 to R09 is shifted out from the shift register.


Transfer start
SPDR (transmission buffer)
Bit 15
Bit 0
T15 T14 T13 T12 T11 T10 T09 T08 T07 T06 T05 T04 T03 T02 T01 T00
Copy
Output
T00 T01 T02 T03 T04 T05 T06 T07 T08 T09 T10 T11 T12 T13 T14 T15
Bit 15
Bit 0
Shift register
Transfer end
Input
Shift register
Bit 15
Bit 0
R00 R01 R02 R03 R04 R05 R06 R07 R08 R09 T10 T11 T12 T13 T14 T15
Bit 6 Bit 5
Copy
T15 T14 T13 T12 T11 T10 R09 R08 R07 R06 R05 R04 R03 R02 R01 R00
Bit 15
Bit 10 Bit 9
Bit 0
SPDR (receive buffer)

Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

Figure 17.13 LSB First Transfer (10-Bit Data)

17.4.6 Transmission Buffer Empty/Receive Buffer Full Flags

Figure 17.14 shows an example of operation of the RSPI transmission buffer empty flag (SPTEF) and the RSPI receive buffer full flag in the RSPI status register (SPSR). The SPDR access depicted in figure 17.14 indicates the condition of access from the A-DMAC to the RSPI data register (SPDR), where I denotes an idle cycle, W a write cycle, and R a read cycle. In the example of figure 17.14, the RSPI performs an 8-bit serial transfer in which the CPHA bit in the RSPI command register (SPCMD) is 1, and CPOL is 0. The numbers given under the RSPCK waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

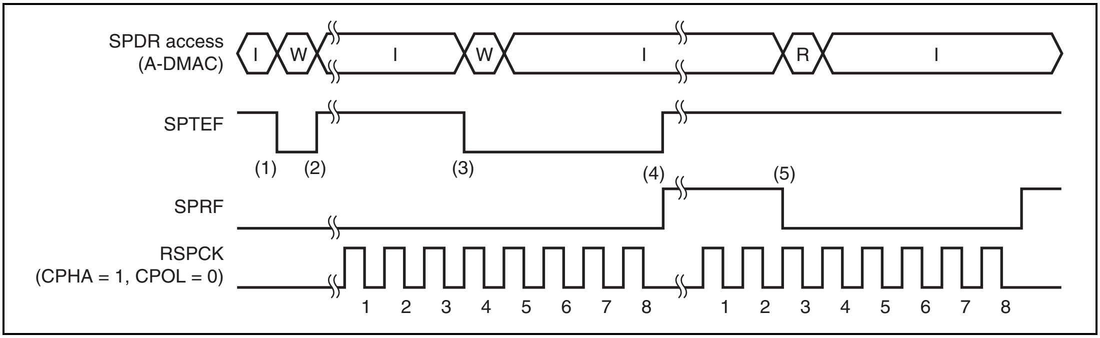


Figure 17.14 SPTEF and SPRF Bit Operation Example

The operation of the flags at timings shown in steps (1) to (5) in the figure is described below.

1. When the A-DMAC writes transmit data to SPDR when the transmission buffer of SPDR is empty, the RSPI sets the SPTEF bit to 0, and writes data to the transmission buffer, with no change in the SPRF flag.
2. If the shift register is empty, the RSPI sets the SPTEF bit to 1, and copies the data in the transmission buffer to the shift register, with no change in the SPRF flag. How a serial transfer is started depends on the mode of the RSPI. For details, see section 17.4.9, Master Mode Operation, and section 17.4.10, Slave Mode Operation.
3. When the A-DMAC writes transmit data to SPDR with the transmission buffer of SPDR being empty, the RSPI sets the SPTEF bit to 1, and writes data to the transmission buffer, while the SPRF flag remains unchanged. Because the data being transferred serially is stored in the shift register, the RSPI does not copy the data in the transmission buffer to the shift register.

4. When the serial transfer ends with the receive buffer of SPDR being empty, the RSPI sets the SPRF bit to 1, and copies the receive data in the shift register to the receive buffer. Because the shift register becomes empty upon completion of serial transfer, if the transmission buffer was full before the serial transfer ended, the RSPI sets the SPTEF bit to 1, and copies the data in the transmission buffer to the shift register. Even when received data is not copied from the shift register to the receive buffer in an overrun error status, upon completion of the serial transfer the RSPI determines that the shift register is empty, and as a result data transfer from the transmission buffer to the shift register is enabled.
5. When the A-DMAC reads SPDR with the receive buffer being full, the RSPI sets the SPRF bit to 0, and sends the data in the receive buffer to the bus inside the chip.

If the CPU or the A-DMAC writes to SPDR when the SPTEF bit is 0, the RSPI does not update the data in the transmission buffer. When writing to SPDR, make sure that the SPTEF bit is 1. That the SPTEF bit is 1 can be checked by reading SPSR or by using an RSPI transmit interrupt. To use an RSPI transmit interrupt, set the SPTIE bit in SPCR to 1.

If the RSPI is disabled (the SPE bit in SPCR being 0), the SPTEF bit is initialized to 1. For this reason, setting the SPTIE bit to 1 when the RSPI is disabled generates an RSPI transmit interrupt.

When serial transfer ends with the SPRF bit being 1, the RSPI does not copy data from the shift register to the receive buffer, and detects an overrun error (see section 17.4.7, Error Detection). To prevent a receive data overrun error, set the SPRF bit to 0 before the serial transfer ends. That the SPRF bit is 1 can be checked by either reading SPSR or by using an RSPI receive interrupt. To use an RSPI receive interrupt, set the SPRIE bit in SPCR to 1.

17.4.7 Error Detection

In the normal RSPI serial transfer, the data written from the RSPI data register (SPDR) to the transmission buffer by either the CPU or the A-DMAC is serially transmitted, and either the CPU or the A-DMAC can read the serially received data from the receive buffer of SPDR. If access is made to SPDR by either the CPU or the A-DMAC, depending on the status of the transmission buffer/receive buffer or the status of the RSPI at the beginning or end of serial transfer, in some cases non-normal transfers can be executed.

If a non-normal transfer operation occurs, the RSPI detects the event as an overrun error or a mode fault error. Table 17.7 shows the relationship between non-normal transfer operations and the RSPI's error detection function.

Table 17.7 Relationship between Non-Normal Transfer Operations and RSPI Error Detection Function

	Occurrence Condition	RSPI Operation	Error Detection
A	Either the CPU or the A-DMAC writes to SPDR when the transmission buffer is full.	Retains the contents of the transmission buffer. Missing write data.	None
B	Serial transfer is started in slave mode when transmit data is still not loaded on the shift register.	Data received in previous serial transfer is serially transmitted.	None
C	Either the CPU or the A-DMAC reads from SPDR when the receive buffer is empty.	Previously received serial data is output to the CPU or the A-DMAC.	None
D	Serial transfer terminates when the receive buffer is full.	Retains the contents of the receive buffer. Missing serial receive data.	Overrun error
E	The SSL0 input signal is asserted when the serial transfer is idle in multi-master mode.	RSPI disabled.	Mode fault error
F	The SSL0 input signal is asserted during serial transfer in multi-master mode.	Serial transfer suspended. Missing send/receive data. Driving of the RSPCK, MOSI, and SSL1 to SSL7 output signals stopped. RSPI disabled.	Mode fault error
G	The SSL0 input signal is negated during serial transfer in slave mode.	Serial transfer suspended. Missing send/receive data. Driving of the MISO output signal stopped. RSPI disabled.	Mode fault error

On operation A shown in table 17.7, the RSPI does not detect an error. To prevent data omission during the writing to SPDR by the CPU or the A-DMAC, write operations to SPDR should be executed when the SPTEF bit in the RSPI status register (SPSR) is 1.

Likewise, the RSPI does not detect an error on operation B. In a serial transfer that was started before the shift register was updated, the RSPI sends the data that was received in the previous serial transfer, and does not treat the operation indicated in B as an error. Notice that the received data from the previous serial transfer is retained in the receive buffer of SPDR, and thus it can be correctly read by the CPU or the A-DMAC (if SPDR is not read before the end of the serial transfer, an overrun error may result).

Similarly, the RSPI does not detect an error on operation C. To prevent the CPU or the A-DMAC from reading extraneous data, SPDR read operation should be executed when the SPRF bit in SPSR is 1.

An overrun error shown in D is described in section 17.4.7 (1), Overrun Error. A mode fault error shown in E to G is described in section 17.4.7 (2), Mode Fault Error. On operations of the SPTEF and SPRF bits in SPSR, see section 17.4.6, Transmission Buffer Empty/Receive Buffer Full Flags.

(1) Overrun Error

If serial transfer ends when the receive buffer of the RSPI data register (SPDR) is full, the RSPI detects an overrun error, and sets the OVRF bit in SPSR to 1. When the OVRF bit is 1, the RSPI does not copy data from the shift register to the receive buffer so that the data prior to the occurrence of the error is retained in the receive buffer. To reset the OVRF bit in SPSR to 0, either execute power-on reset/standby, or write a 0 to the OVRF bit after the CPU has read SPSR with the OVRF bit set to 1.

Figure 17.15 shows an example of operation of the SPRF and OVRF bits in SPSR. The SPSR access depicted in figure 17.15 indicates the condition of access from the CPU to SPSR, and from the A-DMAC to SPDR, respectively, where I denotes an idle cycle, W a write cycle, and R a read cycle. In the example of figure 17.15, the RSPI performs an 8-bit serial transfer in which the CPHA bit in the RSPI command register (SPCMD) is 1, and CPOL is 0. The numbers given under the RSPCK waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

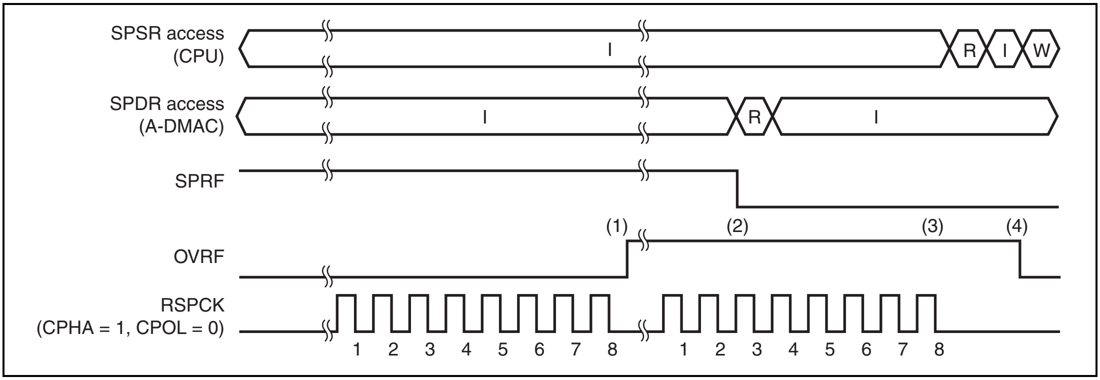


Figure 17.15 SPRF and OVRF Bit Operation Example

The operation of the flags at the timing shown in steps (1) to (4) in the figure is described below.

1. If a serial transfer terminates with the SPRF bit being 1 (receive buffer full), the RSPI detects an overrun error, and sets the OVRF bit to 1. The RSPI does not copy the data in the shift register to the receive buffer. In master mode, the RSPI copies the value of the pointer to the RSPI command register (SPCMD) to bits SPECM2 to SPECM0 in the RSPI sequence status register (SPSSR).
2. When the A-DMAC reads SPDR, the RSPI sets the SPRF bit to 0, and outputs the data in the receive buffer to an internal bus. The receive buffer becoming empty does not clear the OVRF bit.
3. If the serial transfer terminates with the OVRF bit being 1 (an overrun error), the RSPI keeps the SPRF bit at 0 and does not update it. Likewise, the RSPI does not copy the data in the shift register to the receive buffer. When in master mode, the RSPI does not update bits SPECM2 to SPECM0 of SPSSR. If, in an overrun error state, the RSPI does not copy the received data from the shift register to the receive buffer, upon termination of the serial transfer, the RSPI determines that the shift register is empty; in this manner, data transfer is enabled from the transmission buffer to the shift register.
4. If the CPU writes a 0 to the OVRF bit after reading SPSR when the OVRF bit is 1, the RSPI clears the OVRF bit.

The occurrence of an overrun can be checked either by reading SPSR or by using an RSPI error interrupt and reading SPSR. When using an RSPI error interrupt, set the SPEIE bit in the RSPI control register (SPCR) to 1. When executing a serial transfer without using an RSPI error interrupt, measures should be taken to ensure the early detection of overrun errors, such as reading SPSR immediately after SPDR is read. When the RSPI is run in master mode, the pointer value to SPCMD can be checked by reading bits SPECM2 to SPECM0 of SPSSR.

If an overrun error occurs and the OVRF bit is set to 1, normal reception operations cannot be performed until such time as the OVRF bit is cleared. The OVRF bit is cleared to 0 under the following conditions:

- After reading SPSR in a condition in which the OVRF bit is set to 1, the CPU writes a 0 to the OVRF bit.
- Power-on reset/standby

(2) Mode Fault Error

The RSPI operates in multi-master mode when the MSTR bit in the RSPI control register (SPCR) is 1 and the MODFEN bit is also 1. If the active level is input with respect to the SSL0 input signal of the RSPI in multi-master mode, the RSPI detects a mode fault error irrespective of the status of the serial transfer, and sets the MODF bit in the RSPI status register (SPSR) to 1. Upon detecting the mode fault error, the RSPI copies the value of the pointer to the RSPI command register (SPCMD) to bits SPECM2 to SPECM0 in the RSPI sequence status register (SPSSR). The active level of the SSL0 signal is determined by the SSL0P bit in the RSPI slave select polarity register (SSLP).

When the MSTR bit is 0, the RSPI operates in slave mode. The RSPI detects a mode fault error if the MODFEN bit in the RSPI in slave mode is 1, and if the SSL0 input signal is negated during the serial transfer period (from the time the driving of valid data is started to the time the final valid data is fetched).

Upon detecting a mode fault error, the RSPI stops the driving of output signals and clears the SPE bit in the SPCR register. When the SPE bit is cleared, the RSPI function is disabled (see section 17.4.8, Initializing RSPI). In multi-master configuration, it is possible to release the master right by using a mode fault error to stop the driving of output signals and the RSPI function.

The occurrence of a mode fault error can be checked either by reading SPSR or by using an RSPI error interrupt and reading SPSR. When using an RSPI error interrupt, set the SPEIE bit in the RSPI control register (SPCR) to 1. To detect a mode fault error without using an RSPI error interrupt, it is necessary to poll SPSR. When using the RSPI in master mode, one can read bits SPECM2 to SPECM0 of SPSSR to verify the value of the pointer to SPCMD when an error occurs.

When the MODF bit is 1, the RSPI ignores the writing of the value 1 to the SPE bit by the CPU. To enable the RSPI function after the detection of a mode fault error, the MODF bit must be set to 0. The MODF bit is cleared to 0 under the following conditions:

- After reading SPSR in a condition where the MODF bit has turned 1, the CPU writes a 0 to the MODF bit.
- System reset

17.4.8 Initializing RSPI

If the CPU writes a 0 to the SPE bit in the RSPI control register (SPCR) or the RSPI clears the SPE bit to 0 because of the detection of a mode fault error, the RSPI disables the RSPI function, and initializes a part of the module function. During a power-on reset or standby, the RSPI initializes all of the module function. An explanation follows of initialization by the clearing of the SPE bit and initialization by power-on reset/standby.

(1) Initialization by Clearing SPE Bit

When the SPE bit in SPCR is cleared, the RSPI performs the following initialization:

- Suspending any serial transfer that is being executed
- Stopping the driving of output signals only in slave mode (Hi-Z)
- Initializing the internal state of the RSPI
- Initializing the SPTEF bit in the RSPI status register (SPSR)

Initialization by the clearing of the SPE bit does not initialize the control bits of the RSPI. For this reason, the RSPI can be started in the same transfer mode as prior to the initialization if the CPU resets the value 1 to the SPE bit.

The SPRF, OVRF, and MODF bits in SPSR are not initialized, nor is the value of the RSPI sequence status register (SPSSR) initialized. For this reason, even after the RSPI is initialized, data from the receive buffer can be read in order to check the status of error occurrence during an RSPI transfer.

The SPTEF bit in SPSR is initialized to 1. Therefore, if the SPTIE bit in SPCR is set to 1 after RSPI initialization, an RSPI transmit interrupt is generated. When the RSPI is initialized by the CPU, in order to disable any RSPI transmit interrupt, a 0 should be written to the SPTIE bit simultaneously with the writing of a 0 to the SPE bit. To disable any RSPI transmit interrupt after a mode fault error is detected, use an error handling routine to write a 0 to the SPTIE bit.

(2) Power-On Reset/Standby

The initialization by power-on reset/standby completely initializes the RSPI through the initialization of all bits for controlling the RSPI, initialization of the status bits, and initialization of data registers, in addition to the requirements described in section 17.4.8 (1), Initialization by Clearing SPE Bit.

17.4.9 Master Mode Operation

The only difference between single-master mode operation and multi-master mode operation lies in mode fault error detection (see section 17.4.7, Error Detection). When operating in single-master mode, the RSPI does not detect mode fault errors whereas the RSPI running in multi-master mode does detect mode fault errors. This section explains operations that are common to single-/multi-master modes.

(1) Starting Serial Transfer

The RSPI updates the data in the transmission buffer when the SPTEF bit in the RSPI status register (SPSR) is 1 and when either the CPU or the A-DMAC has written data to the RSPI data register (SPDR). If the shift register is empty in a condition where the SPTEF bit has been cleared to 0 due to the writing of 0 either after the writing to SPDR from the A-DMAC or by the writing of 0 after the value 1 is read from the SPTEF bit by the CPU, the RSPI copies the data in the transmission buffer to the shift register and starts a serial transfer. Upon copying transmit data to the shift register, the RSPI changes the status of the shift register to "full", and upon termination of serial transfer, it changes the status of the shift register to "empty". The status of the shift register cannot be referenced from the CPU.

For details on the RSPI transfer format, see section 17.4.4, Transfer Format.

(2) Terminating a Serial Transfer

Irrespective of the CPHA bit in the RSPI command register (SPCMD), the RSPI terminates the serial transfer after transmitting an RSPCK edge corresponding to the final sampling timing. If the SPRF bit in the RSPI status register (SPSR) is 0 and free space is available in the receive buffer, upon termination of serial transfer the RSPI copies data from the shift register to the receive buffer of the RSPI data register (SPDR).

It should be noted that the final sampling timing varies depending on the bit length of transfer data. In master mode, the RSPI data length depends on the settings in bits SPB3 to SPB0 in SPCMD. For details on the RSPI transfer format, see section 17.4.4, Transfer Format.

(3) Sequence Control

The transfer format that is employed in master mode is determined by the RSPI sequence control register (SPSCR), RSPI command registers 0 to 7 (SPCMD0 to SPCMD7), the RSPI bit rate register (SPBR), the RSPI clock delay register (SPCKD), the RSPI slave select negation delay register (SSLND), and the RSPI next-access delay register (SPND).

The SPSCR register is used to determine the sequence configuration for serial transfers that are executed by a master mode RSPI. The following items are set in RSPI command registers SPCMD0 to SPCMD7: SSL output signal value, MSB/LSB first, data length, some of the bit rate settings, RSPCK polarity/phase, whether SPCKD is to be referenced, whether SSLND is to be referenced, and whether SPND is to be referenced. SPBR holds some of the bit rate settings; SPCKD, an RSPI clock delay value; SSLND, an SSL negation delay; and SPND, a next-access delay value.

According to the sequence length that is assigned to SPSCR, the RSPI makes up a sequence comprised of a part or all of SPCMD0 to SPCMD7. The RSPI contains a pointer to the SPCMD that makes up the sequence. The value of this pointer can be checked by reading bits SPCP2 to SPCP0 in the RSPI sequence status register (SPSSR). When the SPE bit in the RSPI control register (SPCR) is set to 1 and the RSPI function is enabled, the RSPI loads the pointer to the commands in SPCMD0, and incorporates the SPCMD0 settings into the transfer format at the beginning of serial transfer. The RSPI increments the pointer each time the next-access delay period for a data transfer ends. Upon completion of the serial transfer that corresponds to the final command comprising the sequence, the RSPI sets the pointer in SPCMD0, and in this manner the sequence is executed repeatedly.

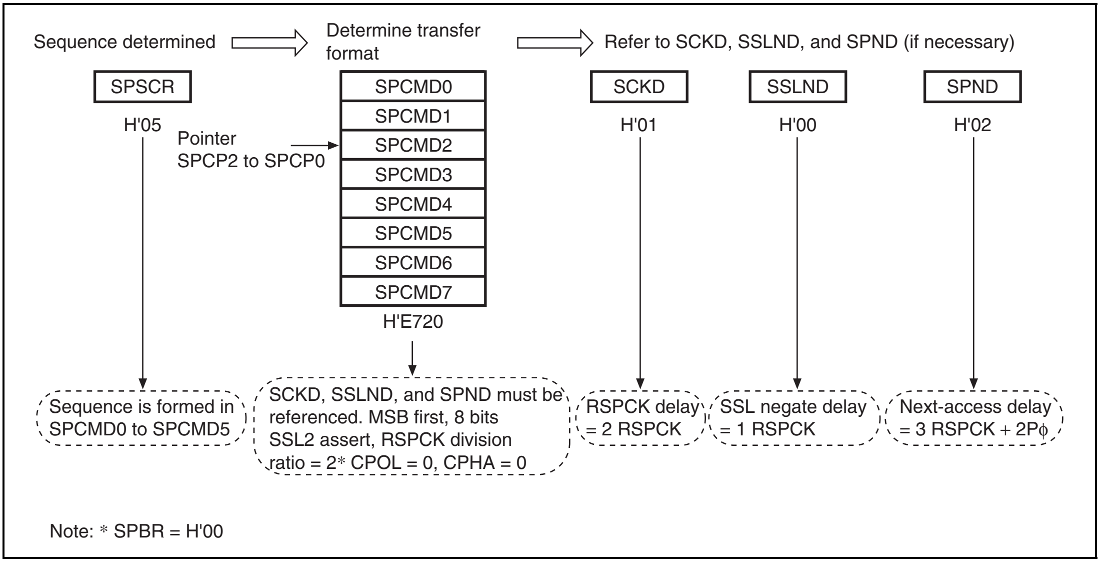

Sequence determined
Determine transfer format
Refer to SCKD, SSLND, and SPND (if necessary)
SPSCR
H'05
Pointer SPCP2 to SPCP0
SPCMD0
SPCMD1
SPCMD2
SPCMD3
SPCMD4
SPCMD5
SPCMD6
SPCMD7
H'E720
SCKD
H'01
SSLND
H'00
SPND
H'02
Sequence is formed in SPCMD0 to SPCMD5
SCKD, SSLND, and SPND must be referenced. MSB first, 8 bits SSL2 assert, RSPCK division ratio = 2* CPOL = 0, CPHA = 0
RSPCK delay = 2 RSPCK
SSL negate delay = 1 RSPCK
Next-access delay = 3 RSPCK + 2Pϕ
Note: * SPBR = H'00

Figure 17.16 Determination Procedure of Serial Transfer Mode in Master Mode

(4) Burst Transfer

If the SSLKP bit in the RSPI command register (SPCMD) that the RSPI references during the current serial transfer is 1, the RSPI keeps the SSL signal level during the serial transfer until the beginning of the SSL signal assertion for the next serial transfer. If the SSL signal level for the next serial transfer is the same as the SSL signal level for the current serial transfer, the RSPI can execute continuous serial transfers while keeping the SSL signal assertion status (burst transfer).

Figure 17.17 shows an example of an SSL signal operation for the case where a burst transfer is implemented using SPCMD0 and SPCMD1 settings. The text below explains the RSPI operations (1) to (7) as depicted in figure 17.17. It should be noted that the polarity of the SSL output signal depends on the settings in the RSPI slave select polarity register (SSLP).

1. Based on SPCMD0, the RSPI asserts the SSL signal and inserts RSPCK delays.
2. The RSPI executes serial transfers according to SPCMD0.
3. The RSPI inserts SSL negation delays.
4. Because the SSLKP bit in SPCMD0 is 1, the RSPI keeps the SSL signal value on SPCMD0. This period is sustained, at the shortest, for a period equal to the next-access delay of SPCMD0. If the shift register is empty after the passage of a minimum period, this period is sustained until such time as the transmit data is stored in the shift register for another transfer.
5. Based on SPCMD1, the RSPI asserts the SSL signal and inserts RSPCK delays.
6. The RSPI executes serial transfers according to SPCMD1.

7. Because the SSLKP bit in SPCMD1 is 0, the RSPI negates the SSL signal. In addition, a next-access delay is inserted according to SPCMD1.

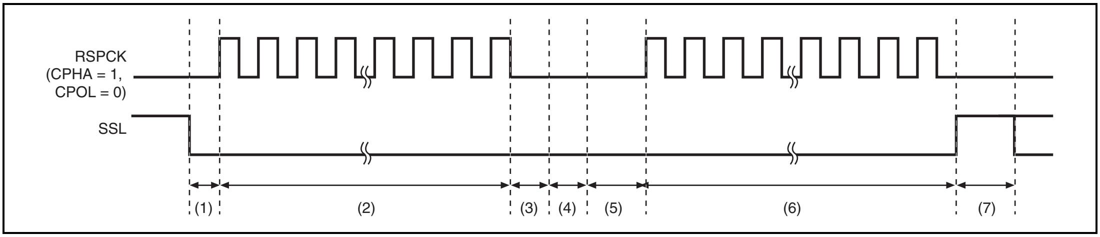


Figure 17.17 Example of Burst Transfer Operation using SSLKP Bit

If the SSL signal settings in the SPCMD in which 1 is assigned to the SSLKP bit are different from the SSL signal output settings in the SPCMD to be used in the next transfer, the RSPI switches the SSL signal status to SSL signal assertion ((5) in figure 17.17) corresponding to the command for the next transfer. Notice that if such an SSL signal switching occurs, the slaves that drive the MISO signal compete, and the possibility arises of the collision of signal levels.

The RSPI in master mode references within the module the SSL signal operation for the case where the SSLKP bit is not used. Even when the CPHA bit in SPCMD is 0, the RSPI can accurately start serial transfers by asserting the SSL signal for the next transfer. For this reason, burst transfers in master mode can be executed irrespective of CPHA bit settings (see section 17.4.10, Slave Mode Operation).

(5) RSPCK Delay (t1)

The RSPCK delay value of the RSPI in master mode depends on SCKDEN bit settings in the RSPI command register (SPCMD) and on RSPCK delay register (SPCKD) settings. The RSPI determines the SPCMD to be referenced during serial transfer by pointer control, and determines an RSPCK delay value during serial transfer by using the SCKDEN bit in the selected SPCMD and SPCKD, as shown in table 17.8. For a definition of RSPCK delay, see section 17.4.4, Transfer Format.

Table 17.8 Relationship among SCKDEN and SPCKD Settings and RSPCK Delay Values

SCKDEN	SPCKD	RSPCK Delay Value
0	000 to 111	1 RSPCK
1	000	1 RSPCK
	001	2 RSPCK
	010	3 RSPCK
	011	4 RSPCK
	100	5 RSPCK
	101	6 RSPCK
	110	7 RSPCK
	111	8 RSPCK

(6) SSL Negation Delay (t2)

The SSL negation delay value of the RSPI in master mode depends on SLNDEN bit settings in the RSPI command register (SPCMD) and on SSL negation delay register (SSLND) settings. The RSPI determines the SPCMD to be referenced during serial transfer by pointer control, and determines an SSL negation delay value during serial transfer by using the SLNDEN bit in the selected SPCMD and SSLND, as shown in table 17.9. For a definition of SSL negation delay, see section 17.4.4, Transfer Format.

Table 17.9 Relationship among SLNDEN and SSLND Settings and SSL Negation Delay Values

SLNDEN	SSLND	SSL Negation Delay Value
0	000 to 111	1 RSPCK
1	000	1 RSPCK
	001	2 RSPCK
	010	3 RSPCK
	011	4 RSPCK
	100	5 RSPCK
	101	6 RSPCK
	110	7 RSPCK
	111	8 RSPCK

(7) Next-Access Delay (t3)

The next-access delay value of the RSPI in master mode depends on SPNDEN bit settings in the RSPI command register (SPCMD) and on next-access delay register (SPND) settings. The RSPI determines the SPCMD to be referenced during serial transfer by pointer control, and determines a next-access delay value during serial transfer by using the SPNDEN bit in the selected SPCMD and SPND, as shown in table 17.10. For a definition of next-access delay, see section 17.4.4, Transfer Format.

Table 17.10 Relationship among SPNDEN and SPND Settings and Next-Access Delay Values

SPNDEN	SPND	Next-Access Delay Value
0	000 to 111	1 RSPCK + 2 Pϕ
1	000	1 RSPCK + 2 Pϕ
	001	2 RSPCK + 2 Pϕ
	010	3 RSPCK + 2 Pϕ
	011	4 RSPCK + 2 Pϕ
	100	5 RSPCK + 2 Pϕ
	101	6 RSPCK + 2 Pϕ
	110	7 RSPCK + 2 Pϕ
	111	8 RSPCK + 2 Pϕ

(8) Initialization Flowchart

Figure 17.18 shows an example of initialization flowchart for using the RSPI in master mode. For a description of how to set up an interrupt controller, the A-DMAC, and input/output ports, see the descriptions given in the individual blocks.

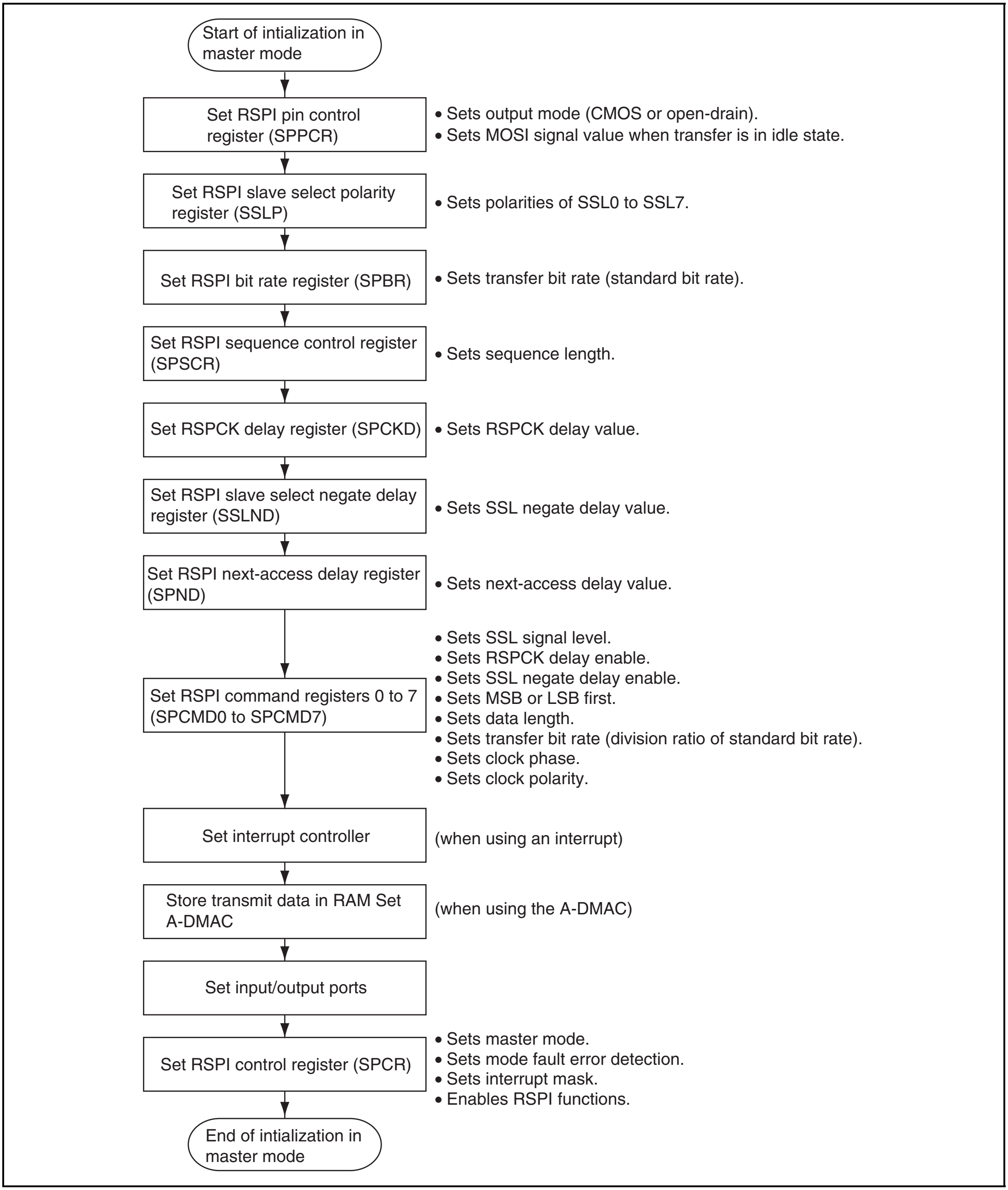
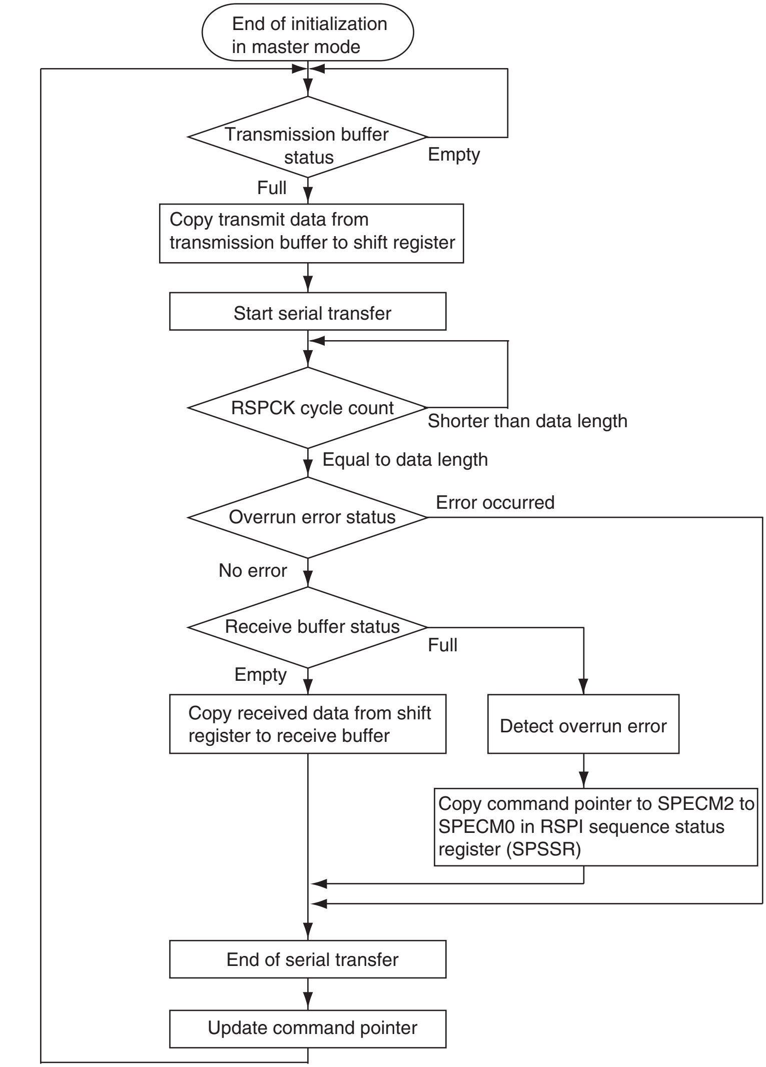

Start of intialization in master mode
Set RSPI pin control register (SPPCR)
• Sets output mode (CMOS or open-drain).
• Sets MOSI signal value when transfer is in idle state.
Set RSPI slave select polarity register (SSLP)
• Sets polarities of SSL0 to SSL7.
Set RSPI bit rate register (SPBR)
• Sets transfer bit rate (standard bit rate).
Set RSPI sequence control register (SPSCR)
• Sets sequence length.
Set RSPCK delay register (SPCKD)
• Sets RSPCK delay value.
Set RSPI slave select negate delay register (SSLND)
• Sets SSL negate delay value.
Set RSPI next-access delay register (SPND)
• Sets next-access delay value.
Set RSPI command registers 0 to 7 (SPCMD0 to SPCMD7)
• Sets SSL signal level.
• Sets RSPCK delay enable.
• Sets SSL negate delay enable.
• Sets MSB or LSB first.
• Sets data length.
• Sets transfer bit rate (division ratio of standard bit rate).
• Sets clock phase.
• Sets clock polarity.
Set interrupt controller
(when using an interrupt)
Store transmit data in RAM Set A-DMAC
(when using the A-DMAC)
Set input/output ports
Set RSPI control register (SPCR)
• Sets master mode.
• Sets mode fault error detection.
• Sets interrupt mask.
• Enables RSPI functions.
End of intialization in master mode

Figure 17.18 Example of Initialization Flowchart in Master Mode

(9) Transfer Operation Flowchart

Figure 17.19 shows an example of transfer operation flowchart for using the RSPI in master mode.


End of initialization in master mode
Transmission buffer status
Empty
Full
Copy transmit data from transmission buffer to shift register
Start serial transfer
RSPCK cycle count
Shorter than data length
Equal to data length
Overrun error status
Error occurred
No error
Receive buffer status
Full
Empty
Copy received data from shift register to receive buffer
Detect overrun error
Copy command pointer to SPECM2 to SPECM0 in RSPI sequence status register (SPSSR)
End of serial transfer
Update command pointer

Note: When the SSL0 input signal is asserted in multi-master mode, the RSPI detects a mode fault error irrespective of the hardware status. When detecting the error, the RSPI copies the command pointer to the SPECM2 to SPECM0 bits in the RSPI sequence status register (SPSSR).

Figure 17.19 Example of Transfer Operation Flowchart in Master Mode

17.4.10 Slave Mode Operation

(1) Starting a Serial Transfer

If the CPHA bit in RSPI command register 0 (SPCMD0) is 0, when detecting an SSL0 input signal assertion, the RSPI needs to start driving valid data to the MISO output signal. For this reason, when the CPHA bit is 0, the asserting of the SSL0 input signal triggers the start of a serial transfer.

If the CPHA bit is 1, when detecting the first RSPCK edge in an SSL0 signal asserted condition, the RSPI needs to start driving valid data to the MSO signal. For this reason, when the CPHA bit is 1, the first RSPCK edge in an SSL0 signal asserted condition triggers the start of a serial transfer.

When detecting the start of a serial transfer in a condition in which the shift register is empty, the RSPI changes the status of the shift register to "full", so that data cannot be copied from the transmission buffer to the shift register when serial transfer is in progress. If the shift register was full before the serial transfer started, the RSPI leaves the status of the shift register intact, in the full state.

Irrespective of CPHA bit settings, the timing at which the RSPI starts driving MISO output signals is the SSL0 signal assertion timing. The data which is output by the RSPI is either valid or invalid, depending on CPHA bit settings.

For details on the RSPI transfer format, see section 17.4.4, Transfer Format. The polarity of the SSL0 input signal depends on the setting of the SSL0P bit in the RSPI slave select polarity register (SSLP).

(2) Terminating a Serial Transfer

Irrespective of the CPHA bit in RSPI command register 0 (SPCMD0), the RSPI terminates the serial transfer after detecting an RSPCK edge corresponding to the final sampling timing. When the SPRF bit in the RSPI status register (SPSR) is 0 and free space is available in the receive buffer, upon termination of serial transfer the RSPI copies received data from the shift register to the receive buffer of the RSPI data register (SPDR). Irrespective of the value of the SPRF bit, upon termination of a serial transfer the RSPI changes the status of the shift register to "empty". When the MODFEN bit in the RSPI control register (SPCR) is 1, a mode fault error occurs if the RSPI detects an SSL0 input signal negation from the beginning of serial transfer to the end of serial transfer (see section 17.4.7, Error Detection).

The final sampling timing changes depending on the bit length of the transfer data. In slave mode, the RSPI data length depends on the settings in bits SPB3 to SPB0 bits in SPCMD0. The polarity of the SSL0 input signal depends on the setting in the SSL0P bit in the RSPI slave select polarity register (SSLP). For details on the RSPI transfer format, see section 17.4.4, Transfer Format.

(3) Notes on Single-Slave Operations

If the CPHA bit in RSPI command register 0(SPCMD0) is 0, the RSPI starts serial transfers when it detects the assertion edge for an SSL0 input signal. In the type of configuration shown in figure 17.4 as an example, if the RSPI is used in single-slave mode, the SSL0 signal is always fixed at active state. Therefore, when the CPHA bit is set to 0, the RSPI cannot correctly start a serial transfer. To correctly execute send/receive operation by the RSPI in a configuration in which the SSL0 input signal is fixed at active state, the CPHA bit should be set to 1. If there is a need for setting the CPHA bit to 0, the SSL0 input signal should not be fixed.

(4) Burst Transfer

If the CPHA bit in RSPI command register 0 (SPCMD0) is 1, continuous serial transfer (burst transfer) can be executed while retaining the assertion state for the SSL0 input signal. If the CPHA bit is 1, the period from the first RSPCK edge to the sampling timing for the reception of the final bit in an SSL0 signal active state corresponds to a serial transfer period. Even when the SSL0 input signal remains at the active level, the RSPI can accommodate burst transfers because it can detect the start of access.

If the CPHA bit is 0, for the reason given in section 17.4.10 (3), Notes on Single-Slave Operations, second and subsequent serial transfers during the burst transfer cannot be executed correctly.

(5) Initialization Flowchart

Figure 17.20 shows an example of initialization flowchart for using the RSPI in slave mode. For a description of how to set up an interrupt controller, the A-DMAC, and input/output ports, see the descriptions given in the individual blocks.

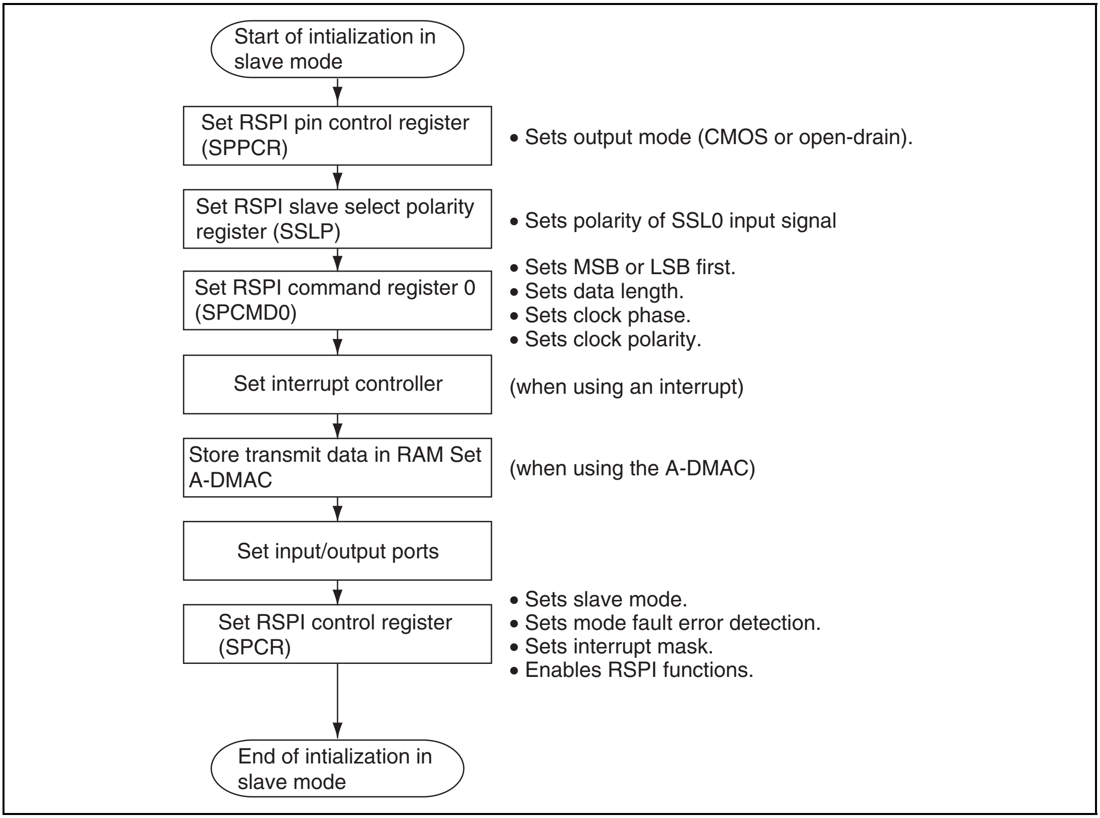

Start of intialization in slave mode
Set RSPI pin control register (SPPCR)
• Sets output mode (CMOS or open-drain).
Set RSPI slave select polarity register (SSLP)
• Sets polarity of SSL0 input signal
Set RSPI command register 0 (SPCMD0)
• Sets MSB or LSB first.
• Sets data length.
• Sets clock phase.
• Sets clock polarity.
Set interrupt controller
(when using an interrupt)
Store transmit data in RAM Set A-DMAC
(when using the A-DMAC)
Set input/output ports
Set RSPI control register (SPCR)
• Sets slave mode.
• Sets mode fault error detection.
• Sets interrupt mask.
• Enables RSPI functions.
End of intialization in slave mode

Figure 17.20 Example of Initialization Flowchart in Slave Mode

(6) Transfer Operation Flowchart (CPHA = 0)

Figure 17.21 shows an example of transfer operation flowchart for using the RSPI in slave mode, when the CPHA bit in RSPI command register 0 (SPCMD0) is 0 and the MODFEN bit in the RSPI control register (SPCR) is 1. When the serial transfer starts while the MODFEN bit is 0, and the SSL0 input level is negated while the RSPCK cycle count is shorter than the data length, operation cannot be guaranteed.

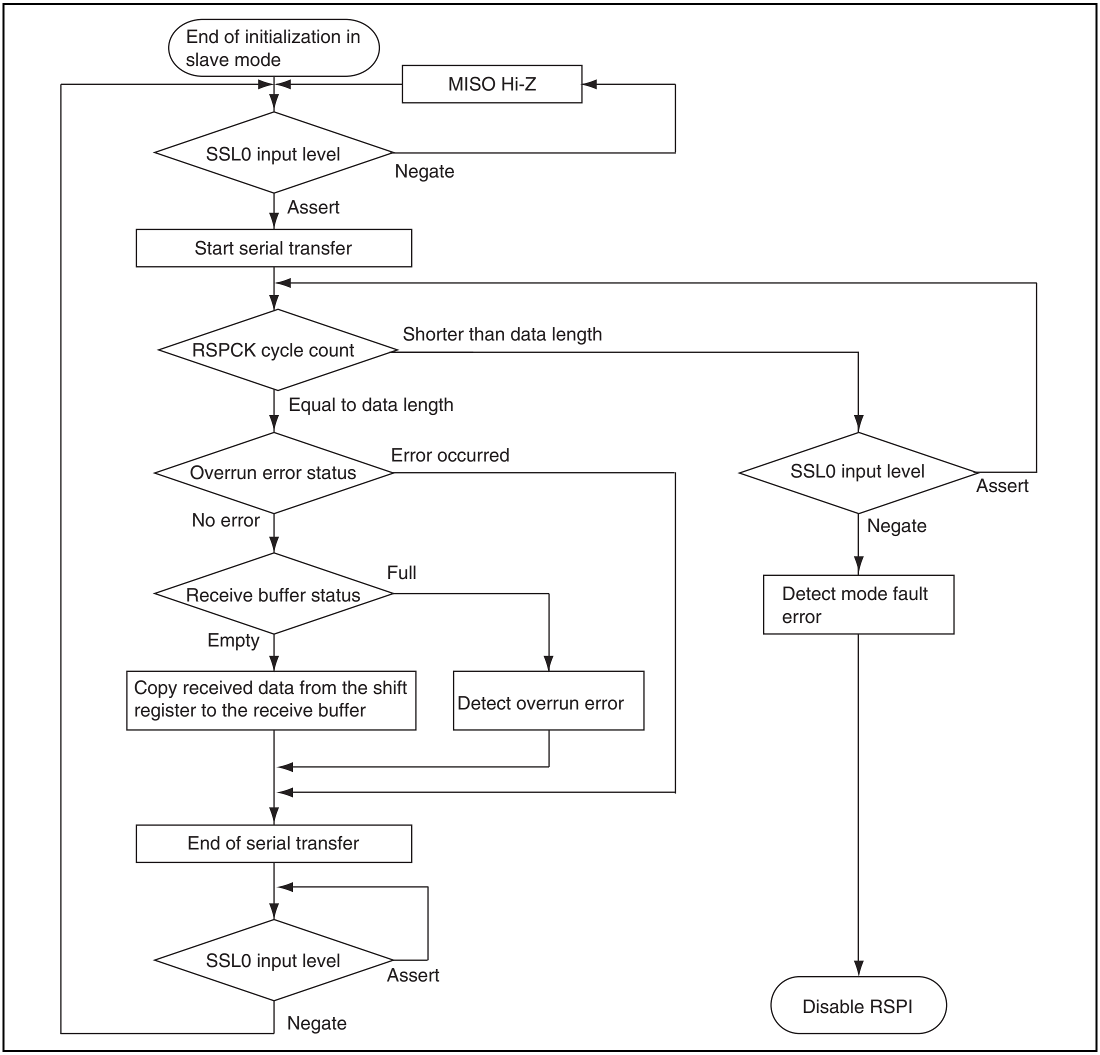


Figure 17.21 Example of Transfer Operation Flowchart in Slave Mode (CPHA = 0, MODFEN = 1)

(7) Transfer Operation Flowchart (CPHA = 1)

Figure 17.22 shows an example of transfer operation flowchart for using the RSPI in slave mode, when the CPHA bit in RSPI command register 0 (SPCMD0) is 1 and the MODFEN bit in the RSPI control register (SPCR) is 1. When the serial transfer starts while the MODFEN bit is 0, and the SSL0 input level is negated while the RSPCK cycle count is shorter than the data length, operation cannot be guaranteed.

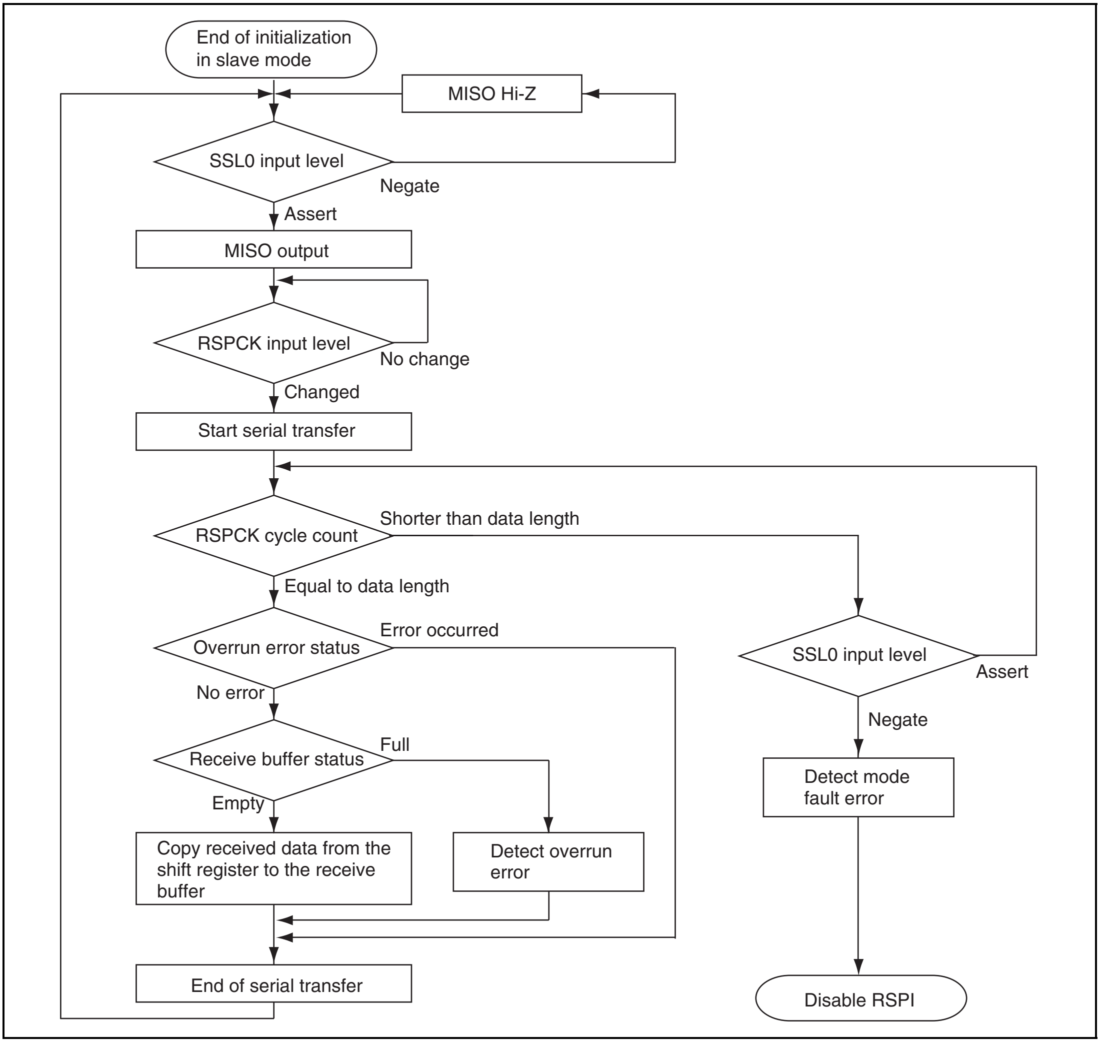


Figure 17.22 Example of Transfer Operation Flowchart in Slave Mode (CPHA = 1, MODFEN = 1)

17.4.11 Loopback Mode

When the CPU writes 1 to the SPLP bit in the RSPI pin control register (SPPCR), the RSPI shuts off the path between the MISO pin and the shift register, and between the MOSI pin and the shift register, and connects the input path and the output path (reversed) of the shift register. This is called loopback mode. When a serial transfer is executed in loopback mode, the transmit data for the RSPI becomes the received data for the RSPI. Figure 17.23 shows the configuration of the shift register input/output paths for the case where the RSPI in master mode is set in loopback mode.

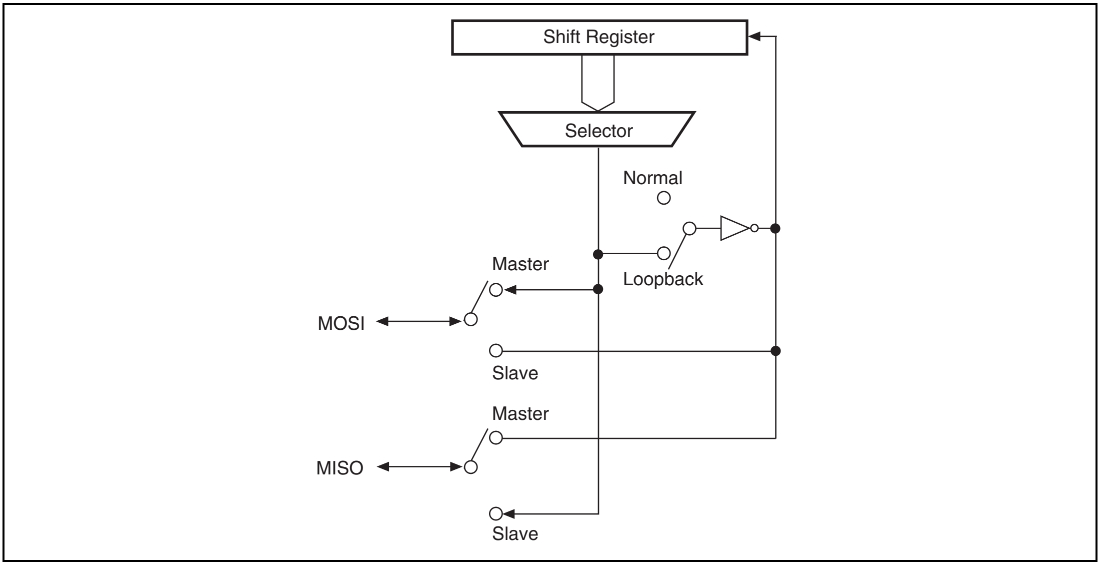


Figure 17.23 Configuration of Shift Register Input/Output Paths in Loopback Mode (Master Mode)

Section 18 Controller Area Network (RCAN-TL1)

18.1 Summary

18.1.1 Overview

This document primarily describes the programming interface for the RCAN-TL1 (Renesas CAN Time Trigger Level 1) module. It serves to facilitate the hardware/software interface so that engineers involved in the RCAN-TL1 implementation can ensure the design is successful.

18.1.2 Scope

The CAN Data Link Controller function is not described in this document. It is the responsibility of the reader to investigate the CAN Specification Document (see references). The interfaces from the CAN Controller are described, in so far as they pertain to the connection with the User Interface.

The programming model is described in some detail. It is not the intention of this document to describe the implementation of the programming interface, but to simply present the interface to the underlying CAN functionality.

The document places no constraints upon the implementation of the RCAN-TL1 module in terms of process, packaging or power supply criteria. These issues are resolved where appropriate in implementation specifications.

18.1.3 Audience

In particular this document provides the design reference for software authors who are responsible for creating a CAN application using this module.

In the creation of the RCAN-TL1 user interface LSI engineers must use this document to understand the hardware requirements.

18.1.4 References

1. CAN Specification Version 2.0 part A, Robert Bosch GmbH, 1991
2. CAN Specification Version 2.0 part B, Robert Bosch GmbH, 1991
3. Implementation Guide for the CAN Protocol, CAN Specification 2.0 Addendum, CAN In Automation, Erlangen, Germany, 1997

4. Road vehicles - Controller area network (CAN): Part 1: Data link layer and physical signalling (ISO-11898-1, 2003)
5. Road vehicles - Controller area network (CAN): Part 4: Time triggered communication (ISO-11898-4, 2004)

18.1.5 Features

- Supports CAN specification 2.0B
- Bit timing compliant with ISO-11898-1
- 32 Mailbox version
- Clock: 16 to 20 MHz, 32 to 40 MHz
- 31 programmable Mailboxes for transmit / receive + 1 receive-only mailbox
- Sleep mode for low power consumption and automatic recovery from sleep mode by detecting CAN bus activity
- Programmable receive filter mask (standard and extended identifier) supported by all Mailboxes
- Programmable CAN data rate up to 1MBit/s
- Transmit message queuing with internal priority sorting mechanism against the problem of priority inversion for real-time applications
- Data buffer access without SW handshake requirement in reception
- Flexible micro-controller interface
- Flexible interrupt structure
- 16-bit free running timer with flexible clock sources and pre-scaler, 3 Timer Compare Match Registers
- 6-bit Basic Cycle Counter for Time Trigger Transmission
- Timer Compare Match Registers with interrupt generation
- Timer counter clear / set capability
- Registers for Time-Trigger: Local_Time, Cycle_time, Ref_Mark, Tx_Enable Window, Ref_Trigger_Offset
- Flexible TimeStamp at SOF for both transmission and reception supported
- Time-Trigger Transmission, Periodic Transmission supported (on top of Event Trigger Transmission)
- Basic Cycle value can be embedded into a CAN frame and transmitted
- Parity: One parity bit is added to every 8 bits of data and parity detection is performed in 32-bit units. A flag is set when a parity error is detected.

18.2 Architecture

The RCAN-TL1 device offers a flexible and sophisticated way to organise and control CAN frames, providing the compliance to CAN2.0B Active and ISO-11898-1. The module is formed from 7 different functional entities. These are the Micro Processor Interface (MPI), Mailbox, Mailbox Control, Timer, CAN Interface, Parity Control, and Parity Circuit. The figure below shows the block diagram of the RCAN-TL1 Module. The bus interface timing is designed according to the peripheral bus I/F required for each product.

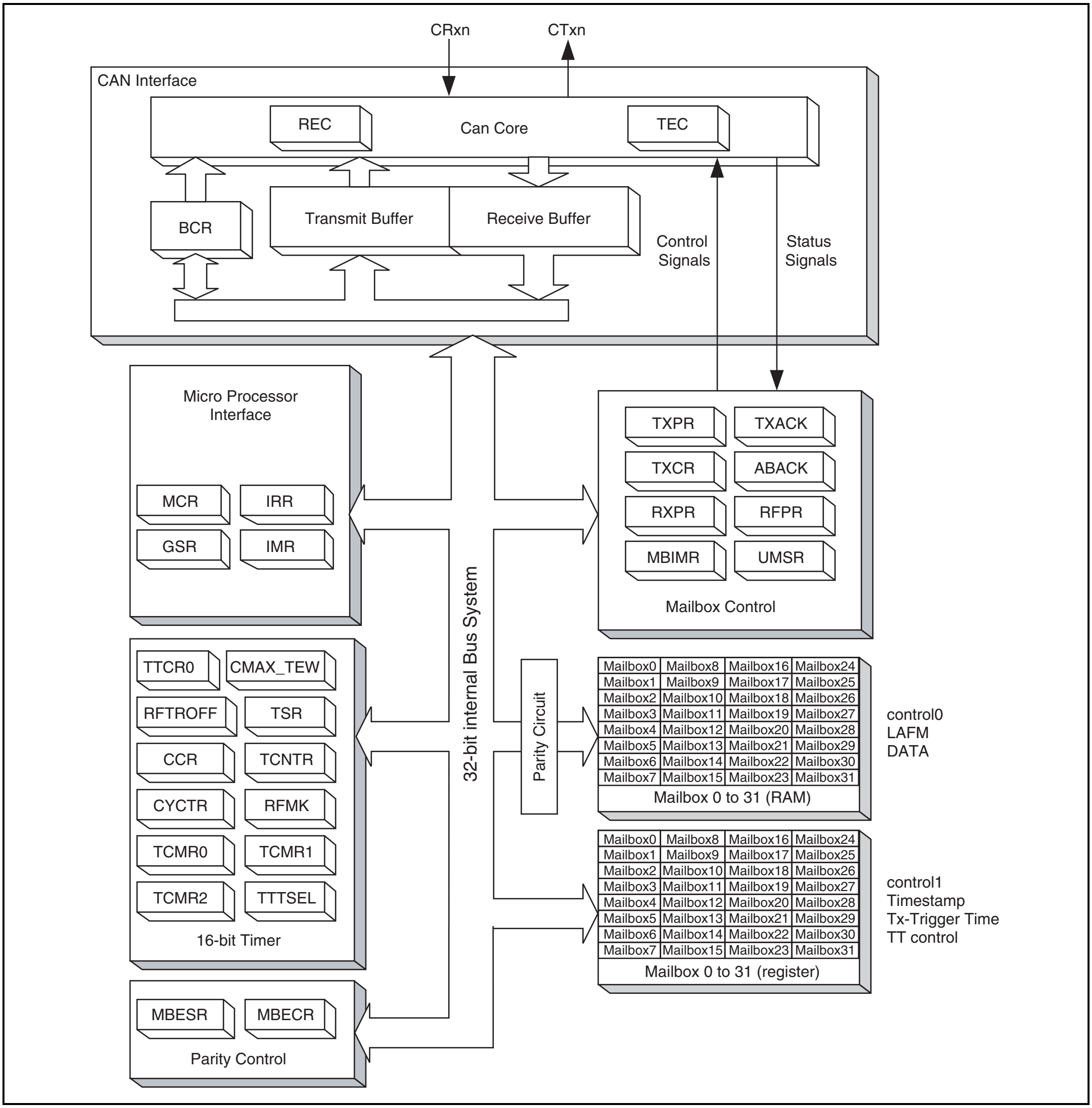

CRxn
CTxn
CAN Interface
REC
Can Core
TEC
BCR
Transmit Buffer
Receive Buffer
Control Signals
Status Signals
Micro Processor Interface
MCR
IRR
GSR
IMR
TXPR
TXACK
TXCR
ABACK
RXPR
RFPR
MBIMR
UMSR
Mailbox Control
32-bit internal Bus System
TTCR0
CMAX_TEW
RFTROFF
TSR
CCR
TCNTR
CYCTR
RFMK
TCMR0
TCMR1
TCMR2
TTTSEL
16-bit Timer
Parity Circuit
Mailbox0 Mailbox8 Mailbox16 Mailbox24
Mailbox1 Mailbox9 Mailbox17 Mailbox25
Mailbox2 Mailbox10 Mailbox18 Mailbox26
Mailbox3 Mailbox11 Mailbox19 Mailbox27
Mailbox4 Mailbox12 Mailbox20 Mailbox28
Mailbox5 Mailbox13 Mailbox21 Mailbox29
Mailbox6 Mailbox14 Mailbox22 Mailbox30
Mailbox7 Mailbox15 Mailbox23 Mailbox31
Mailbox 0 to 31 (RAM)
control0
LAFM
DATA
Mailbox0 Mailbox8 Mailbox16 Mailbox24
Mailbox1 Mailbox9 Mailbox17 Mailbox25
Mailbox2 Mailbox10 Mailbox18 Mailbox26
Mailbox3 Mailbox11 Mailbox19 Mailbox27
Mailbox4 Mailbox12 Mailbox20 Mailbox28
Mailbox5 Mailbox13 Mailbox21 Mailbox29
Mailbox6 Mailbox14 Mailbox22 Mailbox30
Mailbox7 Mailbox15 Mailbox23 Mailbox31
Mailbox 0 to 31 (register)
control1
Timestamp
Tx-Trigger Time
TT control
MBESR
MBECR
Parity Control

Figure 18.1 RCAN-TL1 architecture

Important: Although core of RCAN-TL1 is designed based on a 32-bit bus system, the whole RCAN-TL1 including MPI for the CPU has 16-bit bus interface to CPU. In that case, LongWord (32-bit) access must be implemented as 2 consecutive word (16-bit) accesses. In this manual, LongWord access means the two consecutive accesses.

- Micro Processor Interface (MPI)

  The MPI allows communication between the Renesas CPU and the RCAN-TL1's registers/mailboxes to control the memory interface. It also contains the Wakeup Control logic that detects the CAN bus activities and notifies the MPI and the other parts of RCAN-TL1 so that the RCAN-TL1 can automatically exit the Sleep mode.

  It contains registers such as MCR, IRR, GSR and IMR.

- Mailbox

  The Mailboxes consists of RAM configured as message buffers and registers. There are 32 Mailboxes, and each mailbox has the following information.

  <RAM>

  — CAN message control (identifier, rtr, ide,etc)

  — CAN message data (for CAN Data frames)

  — Local Acceptance Filter Mask for reception

  <Registers>

  — CAN message control (dlc)

  — Time Stamp for message reception/transmission

  — 3-bit wide Mailbox Configuration, Disable Automatic Re-Transmission bit, Auto-Transmission for Remote Request bit, New Message Control bit

  — Tx-Trigger Time

- Mailbox Control

  The Mailbox Control handles the following functions.

  — For received messages, compare the IDs and generate appropriate RAM addresses/data to store messages from the CAN Interface into the Mailbox and set/clear appropriate registers accordingly.

  — To transmit event-triggered messages, run the internal arbitration to pick the correct priority message, and load the message from the Mailbox into the Tx-buffer of the CAN Interface and set/clear appropriate registers accordingly. In the case of time-triggered transmission, compare match of Tx-Trigger time invoke loading the messages.

  — Arbitrates Mailbox accesses between the CPU and the Mailbox Control.

  — Contains registers such as TXPR, TXCR, TXACK, ABACK, RXPR, RFPR, UMSR and MBIMR.

- Timer

  The Timer function is the functional entity, which provides RCAN-TL1 with support for transmitting messages at a specific time frame and recording the result.

  The Timer is a 16-bit free running up counter which can be controlled by the CPU. It provides one 16-bit Compare Match Register to compare with Local Time and two 16-bit ones to compare with Cycle Time. The Compare Match Registers can generate interrupt signals and clear the Counter.

  The clock period of this Timer offers a wide selection derived from the system clock or can be programmed to be incremented with one nominal bit timing of CAN Bus.

  Contains registers such as TCNTR, TTCR0, CMAX_TEW, RETROFF, TSR, CCR, CYCTR, RFMK, TCMR0, TCMR1, TCMR2 and TTTSEL.

- CAN Interface

  This block conforms to the requirements for a CAN Bus Data Link Controller which is specified in Ref. [2, 4]. It fulfils all the functions of a standard DLC as specified by the OSI 7 Layer Reference model. This functional entity also provides the registers and the logic which are specific to a given CAN bus, which includes the Receive Error Counter, Transmit Error Counter, the Bit Configuration Registers and various useful Test Modes. This block also contains functional entities to hold the data received and the data to be transmitted for the CAN Data Link Controller.

- Parity Control

  The Parity Control block allows communications between the CPU and the two parity control registers, MBESR and MBECR.

- Parity Circuit

  Parity bits are added to the data written to a mailbox (RAM). One parity bit is added to every 8 bits of the data.

  Parity check is performed to the data read from a mailbox (RAM), in 32-bit units.

  All mailboxes (RAM) must be initialized. If any data is read from a mailbox (RAM) without initialization, a parity error may occur.

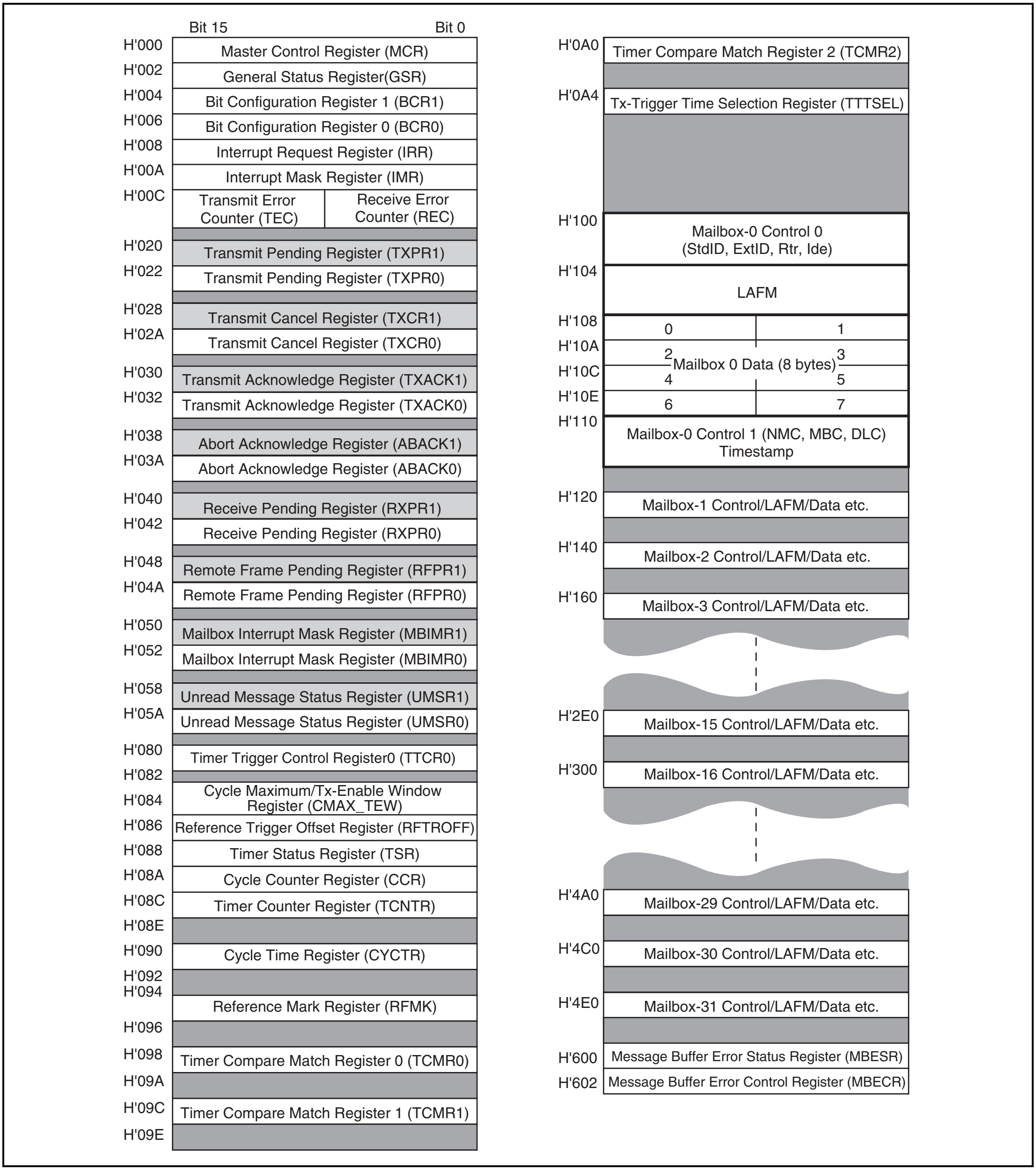
  Generation of a parity error interrupt can be selected by setting the MBECR register.

18.3 Programming Model - Overview

The purpose of this programming interface is to allow convenient, effective access to the CAN bus for efficient message transfer. Please bear in mind that the user manual reports all settings allowed by the RCAN-TL1 IP. Different use of RCAN-TL1 is not allowed.

18.3.1 Memory Map

The diagram of the memory map is shown below.



	Bit 15	Bit 0
H'000	Master Control Register (MCR)	
H'002	General Status Register(GSR)	
H'004	Bit Configuration Register 1 (BCR1)	
H'006	Bit Configuration Register 0 (BCR0)	
H'008	Interrupt Request Register (IRR)	
H'00A	Interrupt Mask Register (IMR)	
H'00C	Transmit Error Counter (TEC)	Receive Error Counter (REC)
H'020	Transmit Pending Register (TXPR1)	
H'022	Transmit Pending Register (TXPR0)	
H'028	Transmit Cancel Register (TXCR1)	
H'02A	Transmit Cancel Register (TXCR0)	
H'030	Transmit Acknowledge Register (TXACK1)	
H'032	Transmit Acknowledge Register (TXACK0)	
H'038	Abort Acknowledge Register (ABACK1)	
H'03A	Abort Acknowledge Register (ABACK0)	
H'040	Receive Pending Register (RXPR1)	
H'042	Receive Pending Register (RXPR0)	
H'048	Remote Frame Pending Register (RFPR1)	
H'04A	Remote Frame Pending Register (RFPR0)	
H'050	Mailbox Interrupt Mask Register (MBIMR1)	
H'052	Mailbox Interrupt Mask Register (MBIMR0)	
H'058	Unread Message Status Register (UMSR1)	
H'05A	Unread Message Status Register (UMSR0)	
H'080	Timer Trigger Control Register0 (TTCR0)	
H'082		
H'084	Cycle Maximum/Tx-Enable Window Register (CMAX_TEW)	
H'086	Reference Trigger Offset Register (RFTROFF)	
H'088	Timer Status Register (TSR)	
H'08A	Cycle Counter Register (CCR)	
H'08C	Timer Counter Register (TCNTR)	
H'08E		
H'090	Cycle Time Register (CYCTR)	
H'092		
H'094	Reference Mark Register (RFMK)	
H'096		
H'098	Timer Compare Match Register 0 (TCMR0)	
H'09A		
H'09C	Timer Compare Match Register 1 (TCMR1)	
H'09E		

H'0A0	Timer Compare Match Register 2 (TCMR2)	
H'0A4	Tx-Trigger Time Selection Register (TTTSEL)	
H'100	Mailbox-0 Control 0 (StdID, ExtID, Rtr, Ide)	
H'104	LAFM	
H'108	0	1
H'10A	2	3
H'10C	4	5
H'10E	6	7
	Mailbox 0 Data (8 bytes)	
H'110	Mailbox-0 Control 1 (NMC, MBC, DLC) Timestamp	
H'120	Mailbox-1 Control/LAFM/Data etc.	
H'140	Mailbox-2 Control/LAFM/Data etc.	
H'160	Mailbox-3 Control/LAFM/Data etc.	
H'2E0	Mailbox-15 Control/LAFM/Data etc.	
H'300	Mailbox-16 Control/LAFM/Data etc.	
H'4A0	Mailbox-29 Control/LAFM/Data etc.	
H'4C0	Mailbox-30 Control/LAFM/Data etc.	
H'4E0	Mailbox-31 Control/LAFM/Data etc.	
H'600	Message Buffer Error Status Register (MBESR)	
H'602	Message Buffer Error Control Register (MBECR)	

Figure 18.2 RCAN-TL1 Memory Map

The locations not used (between H'000 and H'602) are reserved and cannot be accessed.

18.3.2 Mailbox Structure

Mailboxes play a role as message buffers to transmit/receive CAN frames. Each Mailbox is comprised of 3 identical storage fields that are 1): Message Control, 2): Local Acceptance Filter Mask, 3): Message Data. In addition some Mailboxes contain the following extra Fields: 4): Time Stamp, 5): Time Trigger configuration and 6): Time Trigger Control. The following table shows the address map for the control, LAFM, data, timestamp, Transmission Trigger Time and Time Trigger Control addresses for each mailbox.

	Address						
	Control0	**LAFM**	**Data**	**Control1**	**Time Stamp**	**Trigger Time**	**TT control**
Mailbox	**4 bytes**	**4 bytes**	**8 bytes**	**2 bytes**	**2 bytes**	**2 bytes**	**2 bytes**
0 (Receive Only)	H'100 – H'103	H'104– H'107	H'108 – H'10F	H'110 – H'111	H'112 – H'113	No	No
1	H'120 – H'123	H'124 – H'127	H'128 – H'12F	H'130 – H'131	H'132 – H'133	No	No
2	H'140 – H'143	H'144 – H'147	H'148 – H'14F	H'150 – H'151	H'152 – H'153	No	No
3	H'160 – H'163	H'164 - H'167	H'168 – H'16F	H'170 – H'171	H'172 – H'173	No	No
4	H'180 – H'183	H'184 – H'187	H'188 – H'18F	H'190 – H'191	H'192 – H'193	No	No
5	H'1A0 – H'1A3	H'1A4 – H'1A7	H'1A8 – H'1AF	H'1B0 – H'1B1	H'1B2 – H'1B3	No	No
6	H'1C0 – H'1C3	H'1C4 – H'1C7	H'1C8 – H'1CF	H'1D0 – H'1D1	H'1D2 – H'1D3	No	No
7	H'1E0 – H'1E3	H'1E4 – H'1E7	H'1E8 – H'1EF	H'1F0 – H'1F1	H'1F2 – H'1F3	No	No
8	H'200 – H'203	H'204 – H'207	H'208 – H'20F	H'210 – H'211	H'212 – H'213	No	No
9	H'220 – H'223	H'224 – H'227	H'228 – H'22F	H'230 – H'231	H'232 – H'233	No	No
10	H'240 – H'243	H'244 – H'247	H'248 – H'24F	H'250 – H'251	H'252 – H'253	No	No
11	H'260 – H'263	H'264 – H'267	H'268 – H'26F	H'270 – H'271	H'272 – H'273	No	No
12	H'280 – H'283	H'284 – H'287	H'288 – H'28F	H'290 – H'291	H'292 – H'293	No	No
13	H'2A0 – H'2A3	H'2A4 – H'2A7	H'2A8 – H'2AF	H'2B0 – H'2B1	H'2B2 – H'2B3	No	No
14	H'2C0 – H'2C3	H'2C4 – H'2C7	H'2C8 – H'2CF	H'2D0 – H'2D1	H'2D2 – H'2D3	No	No
15	H'2E0 – H'2E3	H'2E4 – H'2E7	H'2E8 – H'2EF	H'2F0 – H'2F1	H'2F2 – H'2F3	No	No
16	H'300 – H'303	H'304 – H'307	H'308 – H'30F	H'310 – H'311	No	No	No
17	H'320 – H'323	H'324 – H'327	H'328 – H'32F	H'330 – H'331	No	No	No
18	H'340 – H'343	H'344 – H'347	H'348 – H'34F	H'350 – H'351	No	No	No

	Address						
	Control0	**LAFM**	**Data**	**Control1**	**Time Stamp**	**Trigger Time**	**TT control**
Mailbox	**4 bytes**	**4 bytes**	**8 bytes**	**2 bytes**	**2 bytes**	**2 bytes**	**2 bytes**
19	H'360 – H'363	H'364 – H'367	H'368 – H'36F	H'370 – H'371	No	No	No
20	H'380 – H'383	H'384 – H'387	H'388 – H'38F	H'390 – H'391	No	No	No
21	H'3A0 – H'3A3	H'3A4 – H'3A7	H'3A8 – H'3AF	H'3B0 – H'3B1	No	No	No
22	H'3C0 – H'3C3	H'3C4 – H'3C7	H'3C8 – H'3CF	H'3D0 – H'3D1	No	No	No
23	H'3E0 – H'3E3	H'3E4 – H'3E7	H'3E8 – H'3EF	H'3F0 – H'3F1	No	No	No
24	H'400 – H'403	H'404 – H'407	H'408 – H'40F	H'410 – H'411	No	H'414 – H'415	H'416 – H'417
25	H'420 – H'423	H'424 – H'427	H'428 – H'42F	H'430 – H'431	No	H'434 – H'435	H'436 – H'437
26	H'440 – H'443	H'444 – H'447	H'448 – H'44F	H'450 – H'451	No	H'454 – H'455	H'456 – H'457
27	H'460 – H'463	H'464 – H'467	H'468 – H'46F	H'470 – H'471	No	H'474 – H'475	H'476 – H'477
28	H'480 – H'483	H'484 – H'487	H'488 – H'48F	H'490 – H'491	No	H'494 – H'495	H'496 – H'497
29	H'4A0 – H'4A3	H'4A4 – H'4A7	H'4A8 – H'4AF	H'4B0 – H'4B1	No	H'4B4 – H'4B5	H'4B6 – H'4B7
30	H'4C0 – H'4C3	H'4C4 – H'4C7	H'4C8 – H'4CF	H'4D0 – H'4D1	H'4D2 – H'4D3 (Local Time)	H'4D4 – H'4D5	No
31	H'4E0 – H'4E3	H'4E4 – H'4E7	H'4E8 – H'4EF	H'4F0 – H'4F1	H'4F2 – H'4F3 (Local Time)	No	No

Mailbox-0 is a receive-only box, and all the other Mailboxes can operate as both receive and transmit boxes, dependant upon the MBC (Mailbox Configuration) bits in the Message Control. The following diagram shows the structure of a Mailbox in detail.

Table 18.1 Roles of Mailboxes

	Event Trigger		Time Trigger		Remark	
	Tx	Rx	Tx	Rx	TimeStamp	Tx-Trigger Time
MB31	OK	OK	—	time reference reception	available	—
MB30	OK	OK	time reference transmission in time master mode	reception in time slave mode	available	available
MB29 - 24	OK	OK	Setting available	Setting available	—	available
MB23 - 16	OK	OK	— (ET)	Setting available	—	—
MB15 - 1	OK	OK	— (ET)	Setting available	available	—
MB0	—	OK	—	Setting available	available	—

(ET) shows that it works during merged arbitrating window, after completion of time-triggered transmission.

MB0 (reception MB with timestamp)

Byte: 8-bit access, Word: 16-bit access, LW (LongWord) : 32-bit access

Address	Data Bus																Access Size	Field Name
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
H'100	IDE	RTR	0	STDID[10:0]											EXTID[17:16]		Word/LW	Control 0
H'102	EXTID[15:0]																Word	
H'104	IDE_LAFM	0	0	STDID_LAFM[10:0]											EXTID_LAFM[17:16]		Word/LW	LAFM
H'106	EXTID_LAFM[15:0]																Word	
H'108	MSG_DATA_0 (first Rx/Tx Byte)								MSG_DATA_1								Byte/Word/LW	Data
H'10A	MSG_DATA_2								MSG_DATA_3								Byte/Word	
H'10C	MSG_DATA_4								MSG_DATA_5								Byte/Word/LW	
H'10E	MSG_DATA_6								MSG_DATA_7								Byte/Word	
H'110	0	0	NMC	0	0	MBC[2:0]			0	0	0	0	DLC[3:0]				Byte/Word	Control 1
H'112	TimeStamp[15:0] (CYCTR[15:0] or CCR[5:0]/CYCTR[15:6] at SOF)																Word	TimeStamp

MBC[1] is fixed to "1"

MB15 to 1 (MB with timestamp)

Address	Data Bus																Access Size	Field Name
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
H'100 + n*32	IDE	RTR	0	STDID[10:0]											EXTID[17:16]		Word/LW	Control 0
H'102 + n*32	EXTID[15:0]																Word	
H'104 + n*32	IDE_LAFM	0	0	STDID_LAFM[10:0]											EXTID_LAFM[17:16]		Word/LW	LAFM
H'106 + n*32	EXTID_LAFM[15:0]																Word	
H'108 + n*32	MSG_DATA_0 (first Rx/Tx Byte)								MSG_DATA_1								Byte/Word/LW	Data
H'10A + n*32	MSG_DATA_2								MSG_DATA_3								Byte/Word	
H'10C + n*32	MSG_DATA_4								MSG_DATA_5								Byte/Word/LW	
H'10E + n*32	MSG_DATA_6								MSG_DATA_7								Byte/Word	
H'110 + n*32	0	0	NMC	ATX	DART	MBC[2:0]			0	0	0	0	DLC[3:0]				Byte/Word	Control 1
H'112 + n*32	TimeStamp[15:0] (CYCTR[15:0] or CCR[5:0]/CYCTR[15:6] at SOF)																Word	TimeStamp

Legend: n = 1 to 15

Figure 18.3 Mailbox-n Structure

MB23 to 16 (MB without timestamp)

Address	Data Bus																Access Size	Field Name
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
H'100 + n*32	IDE	RTR	0	STDID[10:0]											EXTID[17:16]		Word/LW	Control 0
H'102 + n*32	EXTID[15:0]																Word	
H'104 + n*32	IDE_ LAFM	0	0	STDID_LAFM[10:0]											EXTID_ LAFM[17:16]		Word/LW	LAFM
H'106 + n*32	EXTID_LAFM[15:0]																Word	
H'108 + n*32	MSG_DATA_0 (first Rx/Tx Byte)								MSG_DATA_1								Byte/Word/LW	Data
H'10A + n*32	MSG_DATA_2								MSG_DATA_3								Byte/Word	
H'10C + n*32	MSG_DATA_4								MSG_DATA_5								Byte/Word/LW	
H'10E + n*32	MSG_DATA_6								MSG_DATA_7								Byte/Word	
H'110 + n*32	0	0	NMC	ATX	DART	MBC[2:0]			0	0	0	0	DLC[3:0]				Byte/Word	Control 1

MB29 to 24 (Time-Triggered Transmission in Time Trigger mode)

Address	Data Bus																Access Size	Field Name
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
H'100 + n*32	IDE	RTR	0	STDID[10:0]											EXTID[17:16]		Word/LW	Control 0
H'102 + n*32	EXTID[15:0]																Word	
H'104 + n*32	IDE_ LAFM	0	0	STDID_LAFM[10:0]											EXTID_ LAFM[17:16]		Word/LW	LAFM
H'106 + n*32	EXTID_LAFM[15:0]																Word	
H'108 + n*32	MSG_DATA_0 (first Rx/Tx Byte)								MSG_DATA_1								Byte/Word/LW	Data
H'10A + n*32	MSG_DATA_2								MSG_DATA_3								Byte/Word	
H'10C + n*32	MSG_DATA_4								MSG_DATA_5								Byte/Word/LW	
H'10E + n*32	MSG_DATA_6								MSG_DATA_7								Byte/Word	
H'110 + n*32	0	0	NMC	ATX	DART	MBC[2:0]			0	0	0	0	DLC[3:0]				Byte/Word	Control 1
H'112 + n*32	reserved																-	-
H'114 + n*32	Tx-Triggered Time (TTT)																Word	Trigger Time
H'116 + n*32	TTW[1:0]		offset						0	0	0	0	0	Rep_Factor			Word	TT control

Legend: n = 16 to 29

Figure 18.3 Mailbox-n Structure (continued)

MB30 (Time Reference Transmitssion in Time Trigger mode)

Address	Data Bus																Access Size	Field Name
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
H'100 + n*32	IDE	RTR	0	STDID[10:0]											EXTID[17:16]		Word/LW	Control 0
H'102 + n*32	EXTID[15:0]																Word	
H'104 + n*32	IDE_LAFM	0	0	STDID_LAFM[10:0]											EXTID_LAFM[17:16]		Word/LW	LAFM
H'106 + n*32	EXTID_LAFM[15:0]																Word	
H'108 + n*32	MSG_DATA_0 (first Rx/Tx Byte)								MSG_DATA_1								Byte/Word/LW	Data
H'10A + n*32	MSG_DATA_2								MSG_DATA_3								Byte/Word	
H'10C + n*32	MSG_DATA_4								MSG_DATA_5								Byte/Word/LW	
H'10E + n*32	MSG_DATA_6								MSG_DATA_7								Byte/Word	
H'110 + n*32	0	0	NMC	ATX	DART	MBC[2:0]			0	0	0	0	DLC[3:0]				Byte/Word	Control 1
H'112 + n*32	TimeStamp[15:0] (TCNTR at SOF)																Word	TimeStamp
H'114 + n*32	Tx-Triggered Time (TTT) as Time Reference																Word	Trigger Time

MB31 (Time Reference Reception in Time Trigger mode)

Address	Data Bus																Access Size	Field Name
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
H'100 + n*32	IDE	RTR	0	STDID[10:0]											EXTID[17:16]		Word/LW	Control 0
H'102 + n*32	EXTID[15:0]																Word	
H'104 + n*32	IDE_LAFM	0	0	STDID_LAFM[10:0]											EXTID_LAFM[17:16]		Word/LW	LAFM
H'106 + n*32	EXTID_LAFM[15:0]																Word	
H'108 + n*32	MSG_DATA_0 (first Rx/Tx Byte)								MSG_DATA_1								Byte/Word/LW	Data
H'10A + n*32	MSG_DATA_2								MSG_DATA_3								Byte/Word	
H'10C + n*32	MSG_DATA_4								MSG_DATA_5								Byte/Word/LW	
H'10E + n*32	MSG_DATA_6								MSG_DATA_7								Byte/Word	
H'110 + n*32	0	0	NMC	ATX	DART	MBC[2:0]			0	0	0	0	DLC[3:0]				Byte/Word	Control 1
H'112 + n*32	TimeStamp[15:0] (TCNTR at SOF)																Word	TimeStamp

Legend: n = 30, 31

Figure 18.3 Mailbox-n Structure (continued)

Notes:
1. All bits shadowed in grey are reserved and must be written LOW. The value returned by a read may not always be '0' and should not be relied upon.
2. ATX and DART are not supported by Mailbox-0, and the MBC setting of Mailbox-0 is limited.
3. ID Reorder (MCR15) can change the order of STDID, RTR, IDE and EXTID of both message control and LAFM.

(1) Message Control Field

STDID[10:0]: These bits set the identifier (standard identifier) of data frames and remote frames.

EXTID[17:0]: These bits set the identifier (extended identifier) of data frames and remote frames.

RTR (Remote Transmission Request bit): Used to distinguish between data frames and remote frames. This bit is overwritten by received CAN Frames depending on Data Frames or Remote Frames.

Important: Please note that, when ATX bit is set with the setting MBC = 001(bin), the RTR bit will never be set. When a Remote Frame is received, the CPU can be notified by the corresponding RFPR set or IRR[2] (Remote Frame Receive Interrupt), however, as RCAN-TL1 needs to transmit the current message as a Data Frame, the RTR bit remains unchanged.

Important: In order to support automatic answer to remote frame when MBC = 001 (bin) is used and ATX = 1 the RTR flag must be programmed to zero to allow data frame to be transmitted.

Note: When a Mailbox is configured to send a remote frame request the DLC used for transmission is the one stored into the Mailbox.

RTR	Description
0	Data frame
1	Remote frame

IDE (Identifier Extension bit): Used to distinguish between the standard format and extended format of CAN data frames and remote frames.

IDE	Description
0	Standard format
1	Extended format

- Mailbox-0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	NMC	0	0	MBC[2:0]			0	0	0	0	DLC[3:0]			
Initial value:	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R	R	R/W	R	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Note: MBC[1] of MB0 is always "1".

- Mailbox-31 to 1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	NMC	ATX	DART	MBC[2:0]			0	0	0	0	DLC[3:0]			
Initial value:	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

NMC (New Message Control): When this bit is set to '0', the Mailbox of which the RXPR or RFPR bit is already set does not store the new message but maintains the old one and sets the UMSR correspondent bit. When this bit is set to '1', the Mailbox of which the RXPR or RFPR bit is already set overwrites with the new message and sets the UMSR correspondent bit.

Important: Please note that if a remote frame is overwritten with a data frame or vice versa could be that both RXPR and RFPR flags (together with UMSR) are set for the same Mailbox. In this case the RTR bit within the Mailbox Control Field should be relied upon.

Important: Please note that when the Time Triggered mode is used NMC needs to be set to '1' for Mailbox 31 to allow synchronization with all incoming reference messages even when RXPR[31] is not cleared.

NMC	Description
0	Overrun mode (Initial value)
1	Overwrite mode

ATX (Automatic Transmission of Data Frame): When this bit is set to '1' and a Remote Frame is received into the Mailbox DLC is stored. Then, a Data Frame is transmitted from the same Mailbox using the current contents of the message data and updated DLC by setting the corresponding TXPR automatically. The scheduling of transmission is still governed by ID priority or Mailbox priority as configured with the Message Transmission Priority control bit (MCR.2). In order to use this function, MBC[2:0] needs to be programmed to be '001' (Bin). When a transmission is performed by this function, the DLC (Data Length Code) to be used is the one that has been received. Application needs to guarantee that the DLC of the remote frame correspond to the DLC of the data frame requested.

Important: When ATX is used and MBC = 001 (Bin) the filter for the IDE bit cannot be used since ID of remote frame has to be exactly the same as that of data frame as the reply message.

Important: Please note that, when this function is used, the RTR bit will never be set despite receiving a Remote Frame. When a Remote Frame is received, the CPU will be notified by the corresponding RFPR set, however, as RCAN-TL1 needs to transmit the current message as a Data Frame, the RTR bit remains unchanged.

Important: Please note that in case of overrun condition (UMSR flag set when the Mailbox has its NMC = 0) the message received is discarded. In case a remote frame is causing overrun into a Mailbox configured with ATX = 1, the transmission of the corresponding data frame may be triggered only if the related PFPR flag is cleared by the CPU when the UMSR flag is set. In such case PFPR flag would get set again.

ATX	Description
0	Automatic Transmission of Data Frame disabled (Initial value)
1	Automatic Transmission of Data Frame enabled

DART (Disable Automatic Re-Transmission): When this bit is set, it disables the automatic re-transmission of a message in the event of an error on the CAN bus or an arbitration lost on the CAN bus. In effect, when this function is used, the corresponding TXCR bit is automatically set at the start of transmission. When this bit is set to '0', RCAN-TL1 tries to transmit the message as many times as required until it is successfully transmitted or it is cancelled by the TXCR.

DART	Description
0	Re-transmission enabled (Initial value)
1	Re-Transmission disabled

MBC[2:0] (Mailbox Configuration): These bits configure the nature of each Mailbox as follows. When MBC = 111 (Bin), the Mailbox is inactive, i.e., it does not receive or transmit a message regardless of TXPR or other settings. The MBC = '110', '101' and '100' settings are prohibited. When the MBC is set to any other value, the LAFM field becomes available. Please don't set TXPR when MBC is set as reception as there is no hardware protection, and TXPR will remain set. MBC[1] of Mailbox-0 is fixed to "1" by hardware. This is to ensure that MB0 cannot be configured to transmit Messages.

MBC[2]	MBC[1]	MBC[0]	Data Frame Transmit	Remote Frame Transmit	Data Frame Receive	Remote Frame Receive	Remarks
0	0	0	Yes	Yes	No	No	• Not allowed for Mailbox-0 • Time-Triggered transmission can be used
0	0	1	Yes	Yes	No	Yes	• Can be used with ATX* • Not allowed for Mailbox-0 • LAFM can be used
0	1	0	No	No	Yes	Yes	• Allowed for Mailbox-0 • LAFM can be used
0	1	1	No	No	Yes	No	• Allowed for Mailbox-0 • LAFM can be used
1	0	0	Setting prohibited				
1	0	1	Setting prohibited				
1	1	0	Setting prohibited				
1	1	1	Mailbox inactive (Initial value)				

Notes: * In order to support automatic retransmission, RTR shall be "0" when MBC = 001(bin) and ATX = 1.

When ATX = 1 is used the filter for IDE must not be used.

DLC[3:0] (Data Length Code): These bits encode the number of data bytes from 0,1, 2, … 8 that will be transmitted in a data frame. Please note that when a remote frame request is transmitted the DLC value to be used must be the same as the DLC of the data frame that is requested.

DLC[3]	DLC[2]	DLC[1]	DLC[0]	Description
0	0	0	0	Data Length = 0 bytes (Initial value)
0	0	0	1	Data Length = 1 byte
0	0	1	0	Data Length = 2 bytes
0	0	1	1	Data Length = 3 bytes
0	1	0	0	Data Length = 4 bytes
0	1	0	1	Data Length = 5 bytes
0	1	1	0	Data Length = 6 bytes
0	1	1	1	Data Length = 7 bytes
1	x	x	x	Data Length = 8 bytes

(2) Local Acceptance Filter Mask (LAFM)

This area is used as Local Acceptance Filter Mask (LAFM) for receive boxes.

LAFM: When MBC is set to 001, 010, 011(Bin), this field is used as LAFM Field. It allows a Mailbox to accept more than one identifier. The LAFM is comprised of two 16-bit read/write areas as follows.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
H'104 + n*32	IDE_LAFM	0	0	STDID_LAFM[10:0]											EXTID_LAFM[17:16]		Word/LW	LAFM Field
H'106 + n*32	EXTID_LAFM[15:0]																Word	

Legend: n = 0 to 31 (Mailbox number)

Figure 18.4 Acceptance filter

If a bit is set in the LAFM, then the corresponding bit of a received CAN identifier is ignored when the RCAN-TL1 searches a Mailbox with the matching CAN identifier. If the bit is cleared, then the corresponding bit of a received CAN identifier must match to the STDID/IDE/EXTID set in the mailbox to be stored. The structure of the LAFM is same as the message control in a Mailbox. If this function is not required, it must be filled with '0'.

Important: RCAN-TL1 starts to find a matching identifier from Mailbox-31 down to Mailbox-0. As soon as RCAN-TL1 finds one matching, it stops the search. The message will be stored or not depending on the NMC and RXPR/RFPR flags. This means that, even using LAFM, a received message can only be stored into 1 Mailbox.

Important: When a message is received and a matching Mailbox is found, the whole message is stored into the Mailbox. This means that, if the LAFM is used, the STDID, RTR, IDE and EXTID may differ to the ones originally set as they are updated with the STDID, RTR, IDE and EXTID of the received message.

STD_LAFM[10:0] — Filter mask bits for the CAN base identifier [10:0] bits.

STD_LAFM[10:0]	Description
0	Corresponding STD_ID bit is cared
1	Corresponding STD_ID bit is "don't cared"

EXT_LAFM[17:0] — Filter mask bits for the CAN Extended identifier [17:0] bits.

EXT_LAFM[17:0]	Description
0	Corresponding EXT_ID bit is cared
1	Corresponding EXT_ID bit is "don't cared"

IDE_LAFM — Filter mask bit for the CAN IDE bit.

IDE_LAFM	Description
0	Corresponding IDE bit is cared
1	Corresponding IDE bit is "don't cared"

(3) Message Data Fields

Storage for the CAN message data that is transmitted or received. MSG_DATA[0] corresponds to the first data byte that is transmitted or received. The bit order on the CAN bus is bit 7 through to bit 0.

When CMAX ≠ 3'b111/MBC[30] = 3'b000 and TXPR[30] is set, Mailbox-30 is configured as transmission of time reference. Its DLC must be greater than 0 and its RTR must be zero (as specified for TTCAN Level 1) so that the Cycle_count (CCR register) is embedded in the first byte of the data field instead of MSG_DATA_0[5:0] when this Mailbox starts transmission. This function shall be used when RCAN-TL1 is enabled to work in TTCAN mode to perform a Potential Time Master role to send the Time reference message. MSG_DATA_0[7:6] is still transmitted as stored in the Mailbox. User can set MSG_DATA_0[7] when a Next_is_Gap needs to be transmitted.

Please note that the CCR value is only embedded on the frame transmitted but not stored back into Mailbox 30.

When CMAX!= 3'b111, MBC[31] = 3'b011 and TXPR[31] is cleared, Mailbox-31 is configured as reception of time reference. When a valid reference message is received (DLC > 0) RCAN-TL1 performs internal synchronisation (modifying its RFMK and basic cycle CCR).

MB30 - 31

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
H'108 + n*32	Next_is_Gap/Cycle_Counter (first Rx/Tx Byte)								MSG_DATA_1								Byte/Word/LW	Data
H'10A + n*32	MSG_DATA_2								MSG_DATA_3								Byte/Word	
H'10C + n*32	MSG_DATA_4								MSG_DATA_5								Byte/Word/LW	
H'10E + n*32	MSG_DATA_6								MSG_DATA_7								Byte/Word	

Legend: n = 30, 31 (Mailbox number)

Figure 18.5 Message Data Field

(4) Timestamp

Storage for the Timestamp recorded on messages for transmit/receive. The Timestamp will be a useful function to monitor if messages are received/transmitted within expected schedule.

- Timestamp

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TS15	TS14	TS13	TS12	TS11	TS10	TS9	TS8	TS7	TS6	TS5	TS4	TS3	TS2	TS1	TS0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Message Receive: For received messages of Mailbox-15 to 0, Timestamp always captures the CYCTR (Cycle Time Register) value or Cycle_Counter CCR[5:0] + CYCTR[15:6] value, depending on the programmed value in the bit 14 of TTCR0 (Timer Trigger Control Register 0) at SOF.

For messages received into Mailboxes 30 and 31, Timestamp captures the TCNTR (Timer Counter Register) value at SOF.

Message Transmit: For transmitted messages of Mailbox-15 to 1, Timestamp always captures the CYCTR (Cycle Time Register) value or Cycle_Counter CCR[5:0] + CYCTR[15:6] value, depending on the programmed value in the bit 14 of TTCR0 (Timer Trigger Control Register 0), at SOF.

For messages transmitted from Mailboxes30 and 31, Timestamp captures the TCNTR (Timer Counter Register) value at SOF.

Important: Please note that the TimeStamp is stored in a temporary register. Only after a successful transmission or reception the value is then copied into the related Mailbox field. The TimeStamp may also be updated if the CPU clears RXPR/RFPR at the same time that UMSR is set in overrun, however it can be read properly before clearing RXPR/RFPR.

(5) Tx-Trigger Time (TTT) and Time Trigger control

For Mailbox-29 to 24, when MBC is set to 000 (Bin) in time trigger mode (CMAX ≠ 3'b111), Tx-Trigger Time works as Time_Mark to determine the boundary between time windows. The TTT and TT control are comprised of two 16-bit read/write areas as follows. Mailbox-30 doesn't have TT control and works as Time_Ref.

Mailbox 30 to 24 can be used for reception if not used for transmission in TT mode. However they cannot join the event trigger transmission queue when the TT mode is used.

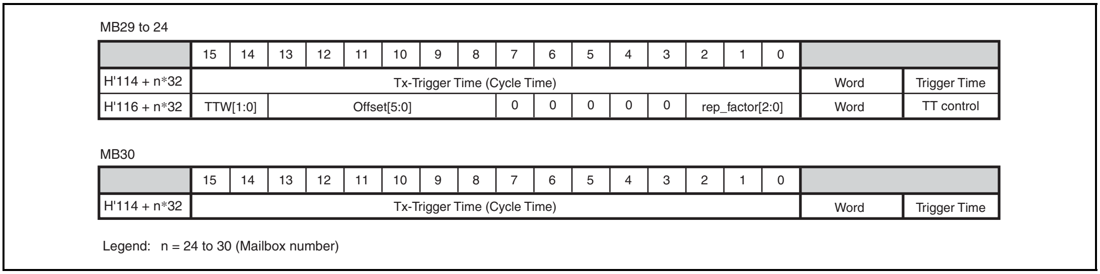
- Tx-Trigger Time

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TTT15	TTT14	TTT13	TTT12	TTT11	TTT10	TTT9	TTT8	TTT7	TTT6	TTT5	TTT4	TTT3	TTT2	TTT1	TTT0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- Time Trigger control

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TTW[1:0]		Offset[5:0]						0	0	0	0	0	rep_factor[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

The following figure shows the differences between all Mailboxes supporting Time Triggered mode.



MB29 to 24

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
H'114 + n*32	Tx-Trigger Time (Cycle Time)																Word	Trigger Time
H'116 + n*32	TTW[1:0]		Offset[5:0]						0	0	0	0	0	rep_factor[2:0]			Word	TT control

MB30

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
H'114 + n*32	Tx-Trigger Time (Cycle Time)																Word	Trigger Time

Legend: n = 24 to 30 (Mailbox number)

Figure 18.6 Tx-Trigger control field

- **TTW[1:0] (Time Trigger Window):** These bits show the attribute of time windows. Please note that once a merged arbitrating window is opened by TTW = 2'b10, the window must be closed by TTW = 2'b11. Several messages with TTW = 2'b10 may be used within the start and the end of a merged arbitrating window.

TTW[1]	TTW[0]	Description
0	0	Exclusive window (initial value)
0	1	Arbitrating window
1	0	Start of merged arbitrating window
1	1	End of merged arbitrating window

The first 16-bit area specifies the time that triggers the transmission of the message in cycle time. The second 16-bit area specifies the basic cycle in the system matrix where the transmission must start (Offset) and the frequency for periodic transmission. When the internal TTT register matches to the CYCTR value, and the internal Offset matches to CCR value transmission is attempted from the corresponding Mailbox. In order to enable this function, the CMAX (Cycle Maximum Register) must be set to a value different from 3'b111, the Timer (TCNTR) must be running (TTCR0 bit15 = 1), the corresponding MBC must be set to 3'b000 and the corresponding TXPR bit must be set. Once TXPR is set by S/W, RCAN-TL1 does not clear the corresponding TXPR bit (among Mailbox-30 to 24) to carry on performing the periodic transmission. In order to stop the periodic transmission, TXPR must be cleared by TXCR. Please note that in this case it is possible that both TXACK and ABACK are set for the same Mailbox if TXACK is not cleared right after completion of transmission. Please refer to Figure 18.7.

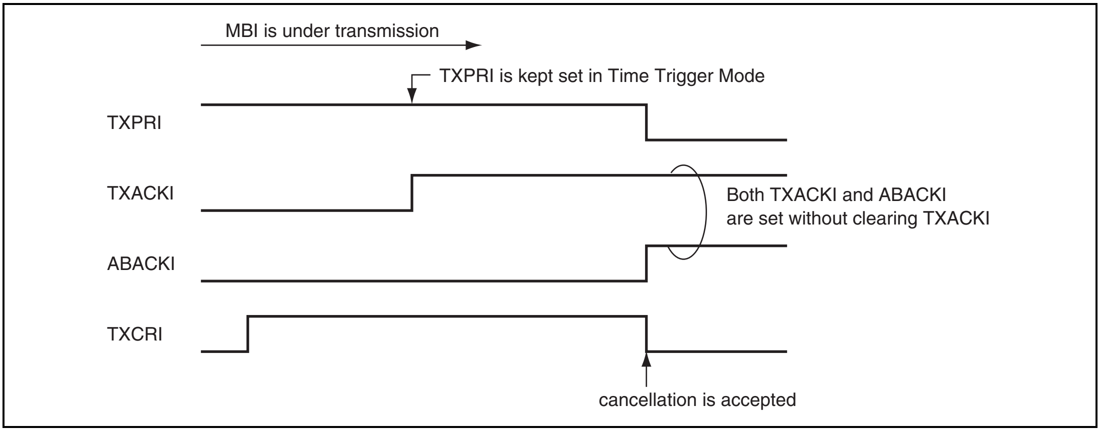

MBI is under transmission
TXPRI is kept set in Time Trigger Mode
TXPRI
TXACKI
Both TXACKI and ABACKI are set without clearing TXACKI
ABACKI
TXCRI
cancellation is accepted

Figure 18.7 TXACK and ABACK in Time Trigger Transmission

Please note that for Mailbox 30 TTW is fixed to '01', Offset to '00' and rep_factor to '0'. The following tables report the combinations for the rep_factor and the offset.

Rep_factor	Description
3'b000	Every basic cycle (initial value)
3'b001	Every two basic cycle
3'b010	Every four basic cycle
3'b011	Every eight basic cycle
3'b100	Every sixteen basic cycle
3'b101	Every thirty two basic cycle
3'b110	Every sixty four basic cycle (once in system matrix)
3'b111	Reserved

The Offset Field determines the first cycle in which a Time Triggered Mailbox may start transmitting its Message.

Offset	Description
6'b000000	Initial Offset = 1st Basic Cycle (initial value)
6'b000001	Initial Offset = 2nd Basic Cycles
6'b000010	Initial Offset = 3rd Basic Cycles
6'b000011	Initial Offset = 4th Basic Cycles
6'b000100	Initial Offset = 5th Basic Cycles
…	
…	
6'b111110	Initial Offset = 63rd Basic Cycles
6'b111111	Initial Offset = 64th Basic Cycles

The following relation must be maintained:

$$\text{Cycle\_Count\_Maximum} + 1 >= \text{Repeat\_Factor} > \text{Offset}$$

$$\text{Cycle\_Count\_Maximum} \quad = 2^{\text{CMAX}} - 1$$

$$\text{Repeat\_Factor} = 2^{\text{rep\_factor}}$$

CMAX, Repeat_Factor, and Offset are register values

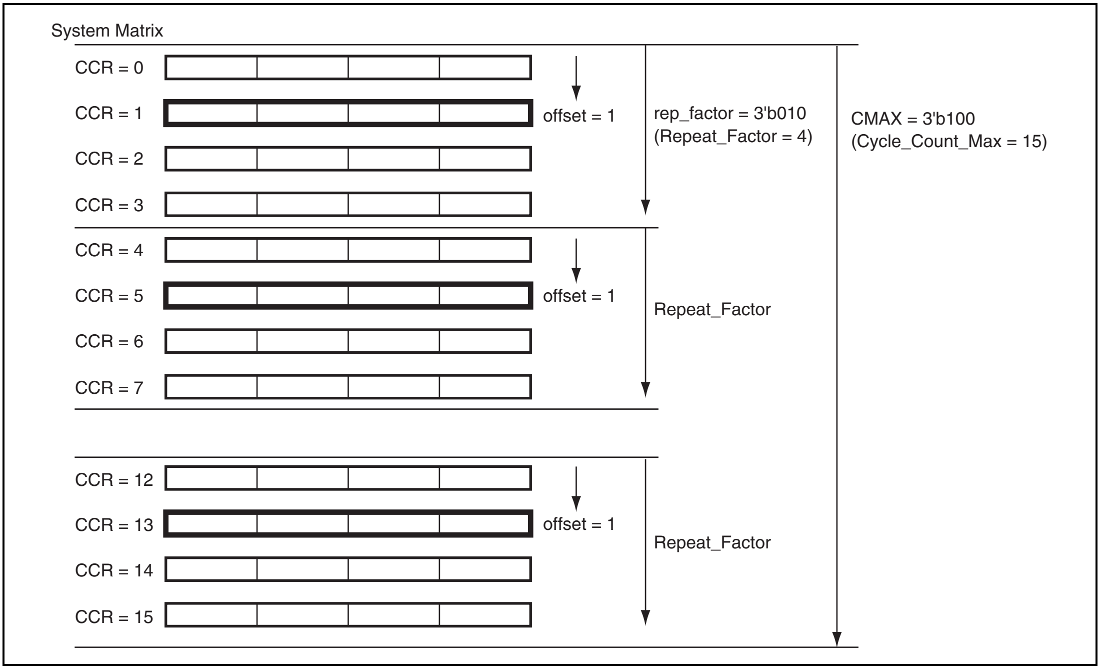


Figure 18.8 System Matrix

Tx-Trigger Times must be set in ascending order such that the difference between them satisfies the following condition.

TTT (mailbox i) - TTT (mailbox i - 1) > TEW + Maximum frame length + 9

18.3.3 RCAN-TL1 Control Registers

The following sections describe RCAN-TL1 control registers. The address is mapped as follow.

Important: These registers can only be accessed in Word size (16-bit).

Description	Address	Name	Access Size (bits)
Master Control Register	000	MCR	Word
General Status Register	002	GSR	Word
Bit Configuration Register 1	004	BCR1	Word
Bit Configuration Register 0	006	BCR0	Word
Interrupt Register	008	IRR	Word
Interrupt Mask Register	00A	IMR	Word
Error Counter Register	00C	TEC/REC	Word
Message Buffer Error Status Register	600	MBESR	Word
Message Buffer Error Control Register	602	MBECR	Word

Figure 18.9 RCAN-TL1 control registers

(1) Master Control Register (MCR)

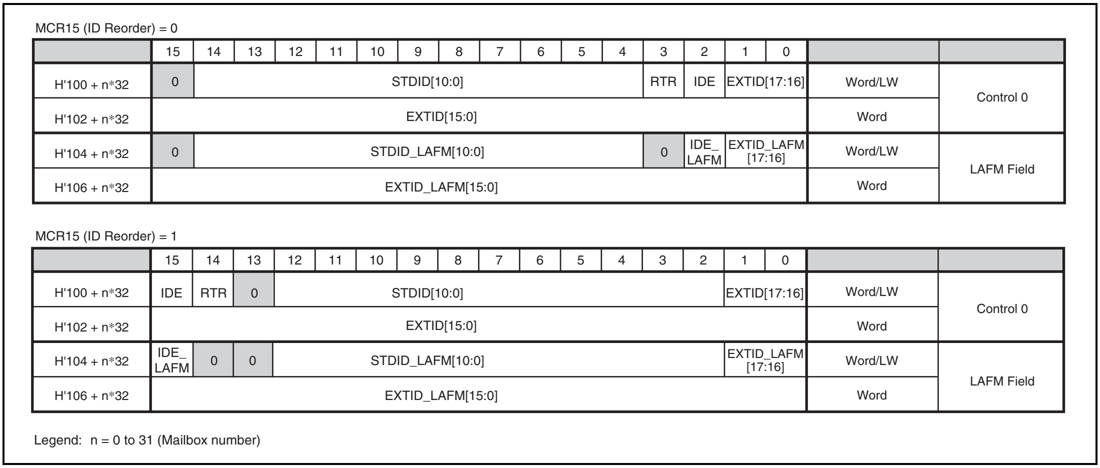
The Master Control Register (MCR) is a 16-bit read/write register that controls RCAN-TL1.

- MCR (Address = H'000)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MCR15	MCR14	-	-	-	TST[2:0]			MCR7	MCR6	MCR5	-	-	MCR2	MCR1	MCR0
Initial value:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W

Bit 15 — ID Reorder (MCR15): This bit changes the order of STDID, RTR, IDE and EXTID of both message control and LAFM.

Bit15: MCR15	Description
0	RCAN-TL1 is the same as HCAN2
1	RCAN-TL1 is not the same as HCAN2 (Initial value)



MCR15 (ID Reorder) = 0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
H'100 + n*32	0	STDID[10:0]											RTR	IDE	EXTID[17:16]		Word/LW	Control 0
H'102 + n*32	EXTID[15:0]																Word	
H'104 + n*32	0	STDID_LAFM[10:0]											0	IDE_ LAFM	EXTID_LAFM [17:16]		Word/LW	LAFM Field
H'106 + n*32	EXTID_LAFM[15:0]																Word	

MCR15 (ID Reorder) = 1

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
H'100 + n*32	IDE	RTR	0	STDID[10:0]											EXTID[17:16]		Word/LW	Control 0
H'102 + n*32	EXTID[15:0]																Word	
H'104 + n*32	IDE_ LAFM	0	0	STDID_LAFM[10:0]											EXTID_LAFM [17:16]		Word/LW	LAFM Field
H'106 + n*32	EXTID_LAFM[15:0]																Word	

Legend: n = 0 to 31 (Mailbox number)

Figure 18.10 ID Reorder

This bit can be modified only in reset mode.

Bit 14 — Auto Halt Bus Off (MCR14): If both this bit and MCR6 are set, MCR1 is automatically set as soon as RCAN-TL1 enters BusOff.

Bit14: MCR14	Description
0	RCAN-TL1 remains in BusOff for normal recovery sequence (128 × 11 Recessive Bits) (Initial value)
1	RCAN-TL1 moves directly into Halt Mode after it enters BusOff if MCR6 is set.

This bit can be modified only in reset mode.

Bit 13 — Reserved. The written value should always be '0' and the returned value is '0'.

Bit 12 — Reserved. The written value should always be '0' and the returned value is '0'.

Bit 11 — Reserved. The written value should always be '0' and the returned value is '0'.

Bit 10 - 8 — Test Mode (TST[2:0]): This bit enables/disables the test modes. Please note that before activating the Test Mode it is requested to move RCAN-TL1 into Halt mode or Reset mode. This is to avoid that the transition to Test Mode could affect a transmission/reception in progress. For details, please refer to section 18.4.1, Test Mode Settings.

Please note that the test modes are allowed only for diagnosis and tests and not when RCAN-TL1 is used in normal operation.

Bit10: TST2	Bit9: TST1	Bit8: TST0	Description
0	0	0	Normal Mode (initial value)
0	0	1	Listen-Only Mode (Receive-Only Mode)
0	1	0	Self Test Mode 1 (External)
0	1	1	Self Test Mode 2 (Internal)
1	0	0	Write Error Counter
1	0	1	Error Passive Mode
1	1	0	Setting prohibited
1	1	1	Setting prohibited

Bit 7 — Auto-wake Mode (MCR7): MCR7 enables or disables the Auto-wake mode. If this bit is set, the RCAN-TL1 automatically cancels the sleep mode (MCR5) by detecting CAN bus activity (dominant bit). If MCR7 is cleared the RCAN-TL1 does not automatically cancel the sleep mode.

RCAN-TL1 cannot store the message that wakes it up.

Note: This bit can be modified only Reset or Halt mode.

Bit7: MCR7	Description
0	Auto-wake by CAN bus activity disabled (Initial value)
1	Auto-wake by CAN bus activity enabled

Bit 6 — Halt during Bus Off (MCR6): MCR6 enables or disables entering Halt mode immediately when MCR1 is set during Bus Off. This bit can be modified only in Reset or Halt mode. Please note that when Halt is entered in Bus Off the CAN engine is also recovering immediately to Error Active mode.

Bit6: MCR6	Description
0	If MCR[1] is set, RCAN-TL1 will not enter Halt mode during Bus Off but wait up to end of recovery sequence (Initial value)
1	Enter Halt mode immediately during Bus Off if MCR[1] or MCR[14] are asserted.

Bit 5 — Sleep Mode (MCR5): Enables or disables Sleep mode transition. If this bit is set, while RCAN-TL1 is in halt mode, the transition to sleep mode is enabled. Setting MCR5 is allowed after entering Halt mode. The two Error Counters (REC, TEC) will remain the same during Sleep mode. This mode will be exited in two ways:

1. by writing a '0' to this bit position,
2. or, if MCR[7] is enabled, after detecting a dominant bit on the CAN bus.

If Auto wake up mode is disabled, RCAN-TL1 will ignore all CAN bus activities until the sleep mode is terminated. When leaving this mode the RCAN-TL1 will synchronise to the CAN bus (by checking for 11 recessive bits) before joining CAN Bus activity. This means that, when the No.2 method is used, RCAN-TL1 will miss the first message to receive. CAN transceivers stand-by mode will also be unable to cope with the first message when exiting stand by mode, and the S/W needs to be designed in this manner.

In sleep mode only the following registers can be accessed: MCR, GSR, IRR and IMR.

Important: RCAN-TL1 is required to be in Halt mode before requesting to enter in Sleep mode. That allows the CPU to clear all pending interrupts before entering sleep mode. Once all interrupts are cleared RCAN-TL1 must leave the Halt mode and enter Sleep mode simultaneously (by writing MCR[5] = 1 and MCR[1] = 0 at the same time).

Bit 5: MCR5	Description
0	RCAN-TL1 sleep mode released (Initial value)
1	Transition to RCAN-TL1 sleep mode enabled

Bit 4 — Reserved. The written value should always be '0' and the returned value is '0'.

Bit 3 — Reserved. The written value should always be '0' and the returned value is '0'.

Bit 2 — Message Transmission Priority (MCR2): MCR2 selects the order of transmission for pending transmit data. If this bit is set, pending transmit data are sent in order of the bit position in the Transmission Pending Register (TXPR). The order of transmission starts from Mailbox-31 as the highest priority, and then down to Mailbox-1 (if those mailboxes are configured for transmission). Please note that this feature cannot be used for time trigger transmission of the Mailboxes 24 to 30.

If MCR2 is cleared, all messages for transmission are queued with respect to their priority (by running internal arbitration). The highest priority message has the Arbitration Field (STDID + IDE bit + EXTID (if IDE = 1) + RTR bit) with the lowest digital value and is transmitted first. The internal arbitration includes the RTR bit and the IDE bit (internal arbitration works in the same way as the arbitration on the CAN Bus between two CAN nodes starting transmission at the same time).

This bit can be modified only in Reset or Halt mode.

Bit 2: MCR2	Description
0	Transmission order determined by message identifier priority (Initial value)
1	Transmission order determined by mailbox number priority (Mailbox-31 → Mailbox-1)

Bit 1—Halt Request (MCR1): Setting the MCR1 bit causes the CAN controller to complete its current operation and then enter Halt mode (where it is cut off from the CAN bus). The RCAN-TL1 remains in Halt Mode until the MCR1 is cleared. During the Halt mode, the CAN Interface does not join the CAN bus activity and does not store messages or transmit messages. All the user registers (including Mailbox contents and TEC/REC) remain unchanged with the exception of IRR0 and GSR4 which are used to notify the halt status itself. If the CAN bus is in idle or intermission state regardless of MCR6, RCAN-TL1 will enter Halt Mode within one Bit Time. If MCR6 is set, a halt request during Bus Off will be also processed within one Bit Time. Otherwise the full Bus Off recovery sequence will be performed beforehand. Entering the Halt Mode can be notified by IRR0 and GSR4.

If both MCR14 and MCR6 are set, MCR1 is automatically set as soon as RCAN-TL1 enters BusOff.

In the Halt mode, the RCAN-TL1 configuration can be modified with the exception of the Bit Timing setting, as it does not join the bus activity. MCR[1] has to be cleared by writing a '0' in order to re-join the CAN bus. After this bit has been cleared, RCAN-TL1 waits until it detects 11 recessive bits, and then joins the CAN bus.

Notes: 1. After issuing a Halt request the CPU is not allowed to set TXPR or TXCR or clear MCR1 until the transition to Halt mode is completed (notified by IRR0 and GSR4). After MCR1 is set this can be cleared only after entering Halt mode or through a reset operation (SW or HW).

2. Transition into or recovery from HALT mode, is only possible if the BCR1 and BCR0 registers are configured to a proper Baud Rate.

Bit 1: MCR1	Description
0	Clear Halt request (Initial value)
1	Halt mode transition request

Bit 0 — Reset Request (MCR0): Controls resetting of the RCAN-TL1 module. When this bit is changed from '0' to '1' the RCAN-TL1 controller enters its reset routine, re-initializing the internal logic, which then sets GSR3 and IRR0 to notify the reset mode. During a re-initialization, all user registers are initialized.

RCAN-TL1 can be re-configured while this bit is set. This bit has to be cleared by writing a '0' to join the CAN bus. After this bit is cleared, the RCAN-TL1 module waits until it detects 11 recessive bits, and then joins the CAN bus. The Baud Rate needs to be set up to a proper value in order to sample the value on the CAN Bus.

After Power On Reset, this bit and GSR3 are always set. This means that a reset request has been made and RCAN-TL1 needs to be configured.

The Reset Request is equivalent to a Power On Reset but controlled by Software.

Bit 0: MCR0	Description
0	Clear Reset Request
1	CAN Interface reset mode transition request (Initial value)

(2) General Status Register (GSR)

The General Status Register (GSR) is a 16-bit read-only register that indicates the status of RCAN-TL1.

- GSR (Address = H'002)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	GSR5	GSR4	GSR3	GSR2	GSR1	GSR0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bits 15 to 6: Reserved. The written value should always be '0' and the returned value is '0'.

Bit 5 — Error Passive Status Bit (GSR5): Indicates whether the CAN Interface is in Error Passive or not. This bit will be set high as soon as the RCAN-TL1 enters the Error Passive state and is cleared when the module enters again the Error Active state (this means the GSR5 will stay high during Error Passive and during Bus Off). Consequently to find out the correct state both GSR5 and GSR0 must be considered.

Bit 5: GSR5	Description
0	RCAN-TL1 is not in Error Passive or in Bus Off status (Initial value) [Reset condition] RCAN-TL1 is in Error Active state
1	RCAN-TL1 is in Error Passive (if GSR0 = 0) or Bus Off (if GSR0 = 1) [Setting condition] When TEC ≥ 128 or REC ≥ 128 or if Error Passive Test Mode is selected

Bit 4 — Halt/Sleep Status Bit (GSR4): Indicates whether the CAN engine is in the halt/sleep state or not. Please note that the clearing time of this flag is not the same as the setting time of IRR12.

Please note that this flag reflects the status of the CAN engine and not of the full RCAN-TL1 IP. RCAN-TL1 exits sleep mode and can be accessed once MCR5 is cleared. The CAN engine exits sleep mode only after two additional transmission clocks on the CAN Bus.

Bit 4: GSR4	Description
0	RCAN-TL1 is not in the Halt state or Sleep state (Initial value)
1	Halt mode (if MCR1 = 1) or Sleep mode (if MCR5 = 1) [Setting condition] If MCR1 is set and the CAN bus is either in intermission or idle or MCR5 is set and RCAN-TL1 is in the halt mode or RCAN-TL1 is moving to Bus Off when MCR14 and MCR6 are both set

Bit 3 — Reset Status Bit (GSR3): Indicates whether the RCAN-TL1 is in the reset state or not.

Bit 3: GSR3	Description
0	RCAN-TL1 is not in the reset state
1	Reset state (Initial value) [Setting condition] After an RCAN-TL1 internal reset (due to SW or HW reset)

Bit 2 — Message Transmission in progress Flag (GSR2): Flag that indicates to the CPU if the RCAN-TL1 is in Bus Off or transmitting a message or an error/overload flag due to error detected during transmission. The timing to set TXACK is different from the time to clear GSR2. TXACK is set at the 7th bit of End Of Frame. GSR2 is set at the 3rd bit of intermission if there are no more messages ready to be transmitted. It is also set by arbitration lost, bus idle, reception, reset or halt transition.

Bit 2: GSR2	Description
0	RCAN-TL1 is in Bus Off or a transmission is in progress
1	[Setting condition] Not in Bus Off and no transmission in progress (Initial value)

Bit 1—Transmit/Receive Warning Flag (GSR1): Flag that indicates an error warning.

Bit 1: GSR1	Description
0	[Reset condition] When (TEC < 96 and REC < 96) or Bus Off (Initial value)
1	[Setting condition] When 96 ≤ TEC < 256 or 96 ≤ REC < 256

Note: REC is incremented during Bus Off to count the recurrences of 11 recessive bits as requested by the Bus Off recovery sequence. However the flag GSR1 is not set in Bus Off.

Bit 0—Bus Off Flag (GSR0): Flag that indicates that RCAN-TL1 is in the bus off state.

Bit 0: GSR0	Description
0	[Reset condition] Recovery from bus off state or after a HW or SW reset (Initial value)
1	[Setting condition] When TEC ≥ 256 (bus off state)

Note: Only the lower 8 bits of TEC are accessible from the user interface. The 9th bit is equivalent to GSR0.

(3) Bit Configuration Register (BCR0, BCR1)

The bit configuration registers (BCR0 and BCR1) are 2 X 16-bit read/write register that are used to set CAN bit timing parameters and the baud rate pre-scaler for the CAN Interface.

The Time quanta is defined as:

$$Timequanta = \frac{2 * BRP}{f_{clk}}$$

Where: BRP (Baud Rate Pre-scaler) is the value stored in BCR0 incremented by 1 and fclk is the used peripheral bus frequency.

- BCR1 (Address = H'004)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSG1[3:0]				-	TSG2[2:0]			-	-	SJW[1:0]		-	-	-	BSP
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R/W	R/W	R	R	R	R/W

Bits 15 to 12 — Time Segment 1 (TSG1[3:0] = BCR1[15:12]): These bits are used to set the segment TSEG1 (= PRSEG + PHSEG1) to compensate for edges on the CAN Bus with a positive phase error. A value from 4 to 16 time quanta can be set.

Bit 15: TSG1[3]	Bit 14: TSG1[2]	Bit 13: TSG1[1]	Bit 12: TSG1[0]	Description
0	0	0	0	Setting prohibited (Initial value)
0	0	0	1	Setting prohibited
0	0	1	0	Setting prohibited
0	0	1	1	PRSEG + PHSEG1 = 4 time quanta
0	1	0	0	PRSEG + PHSEG1 = 5 time quanta
: :	: :	: :	: :	: :
1	1	1	1	PRSEG + PHSEG1 = 16 time quanta

Bit 11: Reserved. The written value should always be '0' and the returned value is '0'.

Bits 10 to 8 — Time Segment 2 (TSG2[2:0] = BCR1[10:8]): These bits are used to set the segment TSEG2 (= PHSEG2) to compensate for edges on the CAN Bus with a negative phase error. A value from 2 to 8 time quanta can be set as shown below.

Bit 10: TSG2[2]	Bit 9: TSG2[1]	Bit 8: TSG2[0]	Description
0	0	0	Setting prohibited (Initial value)
0	0	1	PHSEG2 = 2 time quanta (conditionally prohibited)
0	1	0	PHSEG2 = 3 time quanta
0	1	1	PHSEG2 = 4 time quanta
1	0	0	PHSEG2 = 5 time quanta
1	0	1	PHSEG2 = 6 time quanta
1	1	0	PHSEG2 = 7 time quanta
1	1	1	PHSEG2 = 8 time quanta

Bits 7 and 6: Reserved. The written value should always be '0' and the returned value is '0'.

Bits 5 and 4 - ReSynchronisation Jump Width (SJW[1:0] = BCR0[5:4]): These bits set the synchronisation jump width.

Bit 5: SJW[1]	Bit 4: SJW[0]	Description
0	0	Synchronisation Jump width = 1 time quantum (Initial value)
0	1	Synchronisation Jump width = 2 time quanta
1	0	Synchronisation Jump width = 3 time quanta
1	1	Synchronisation Jump width = 4 time quanta

Bits 3 to 1: Reserved. The written value should always be '0' and the returned value is '0'.

Bit 0 — Bit Sample Point (BSP = BCR1[0]): Sets the point at which data is sampled.

Bit 0 : BSP	Description
0	Bit sampling at one point (end of time segment 1) (Initial value)
1	Bit sampling at three points (rising edge of the last three clock cycles of PHSEG1)

- BCR0 (Address = H'006)

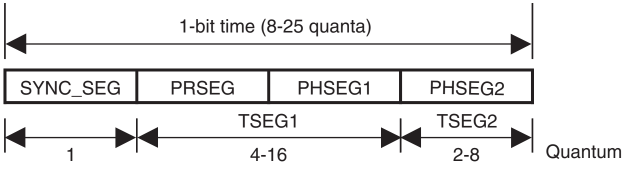
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	BRP[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 8 to 15: Reserved. The written value should always be '0' and the returned value is '0'.

Bits 7 to 0—Baud Rate Pre-scale (BRP[7:0] = BCR0 [7:0]): These bits are used to define the peripheral bus clock periods contained in a Time Quantum.

Bit 7: BRP[7]	Bit 6: BRP[6]	Bit 5: BRP[5]	Bit 4: BRP[4]	Bit 3: BRP[3]	Bit 2: BRP[2]	Bit 1: BRP[1]	Bit 0: BRP[0]	Description
0	0	0	0	0	0	0	0	2 X peripheral bus clock (Initial value)
0	0	0	0	0	0	0	1	4 X peripheral bus clock
0	0	0	0	0	0	1	0	6 X peripheral bus clock
: :	: :	: :	: :	: :	: :	: :	: :	2 X (register value + 1) X peripheral bus clock
1	1	1	1	1	1	1	1	512 X peripheral bus clock

- Requirements of Bit Configuration Register



SYNC_SEG: Segment for establishing synchronisation of nodes on the CAN bus. (Normal bit edge transitions occur in this segment.)

PRSEG: Segment for compensating for physical delay between networks.

PHSEG1: Buffer segment for correcting phase drift (positive). (This segment is extended when synchronisation (resynchronisation) is established.)

PHSEG2: Buffer segment for correcting phase drift (negative). (This segment is shortened when synchronisation (resynchronisation) is established)

TSEG1: TSG1 + 1

TSEG2: TSG2 + 1

The RCAN-TL1 Bit Rate Calculation is:

$$\text{Bit Rate} = \frac{f_{clk}}{2 \times (BRP + 1) \times (TSEG1 + TSEG2 + 1)}$$

Where BRP is given by the register value and TSEG1 and TSEG2 are derived values from TSG1 and TSG2 register values. The '+1' in the above formula is for the Sync-Seg which duration is 1 time quanta.

f_{CLK} = Peripheral Clock

BCR Setting Constraints

TSEG1min > TSEG2 ≥ SJWmax (SJW = 1 to 4)

8 ≤ TSEG1 + TSEG2 + 1 ≤ 25 time quanta (TSEG1 + TSEG2 + 1 = 7 is not allowed)

TSEG2 ≥ 2

These constraints allow the setting range shown in the table below for TSEG1 and TSEG2 in the Bit Configuration Register. The number in the table shows possible setting of SJW. "No" shows that there is no allowed combination of TSEG1 and TSEG2.

		001	010	011	100	101	110	111	TSG2
		2	3	4	5	6	7	8	TSEG2
TSG1	TSEG1								
0011	4	No	1-3	No	No	No	No	No	
0100	5	1-2	1-3	1-4	No	No	No	No	
0101	6	1-2	1-3	1-4	1-4	No	No	No	
0110	7	1-2	1-3	1-4	1-4	1-4	No	No	
0111	8	1-2	1-3	1-4	1-4	1-4	1-4	No	
1000	9	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1001	10	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1010	11	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1011	12	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1100	13	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1101	14	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1110	15	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1111	16	1-2	1-3	1-4	1-4	1-4	1-4	1-4	

Example 1: To have a Bit rate of 500Kbps with a frequency of fclk = 40MHz it is possible to set: BRP = 3, TSEG1 = 6, TSEG2 = 3.

Then the configuration to write is BCR1 = 5200 and BCR0 = 0003.

Example 2: To have a Bit rate of 250Kps with a frequency of 35MHz it is possible to set: BPR = 4, TSEG1 = 8, TSEG2 = 5.

Then the configuration to write is BCR1 = 7400 and BCR0 = 0004.

(4) Interrupt Request Register (IRR)

The interrupt register (IRR) is a 16-bit read/write-clearable register containing status flags for the various interrupt sources.

- IRR (Address = H'008)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IRR15	IRR14	IRR13	IRR12	IRR11	IRR10	IRR9	IRR8	IRR7	IRR6	IRR5	IRR4	IRR3	IRR2	IRR1	IRR0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R/W

Bit 15 — Timer Compare Match Interrupt 1 (IRR15): Indicates that a Compare-Match condition occurred to the Timer Compare Match Register 1 (TCMR1). When the value set in the TCMR1 matches to Cycle Time (TCMR1 = CYCTR), this bit is set.

Bit 15: IRR15	Description
0	Timer Compare Match has not occurred to the TCMR1 (Initial value) [Clearing condition] Writing 1
1	Timer Compare Match has occurred to the TCMR1 [Setting condition] TCMR1 matches to Cycle Time (TCMR1 = CYCTR)

Bit 14 — Timer Compare Match Interrupt 0 (IRR14): Indicates that a Compare-Match condition occurred to the Timer Compare Match Register 0 (TCMR0). When the value set in the TCMR0 matches to Local Time (TCMR0 = TCNTR), this bit is set.

Bit 14: IRR14	Description
0	Timer Compare Match has not occurred to the TCMR0 (Initial value) [Clearing condition] Writing 1
1	Timer Compare Match has occurred to the TCMR0 [Setting condition] TCMR0 matches to the Timer value (TCMR0 = TCNTR)

Bit 13 - Timer Overrun Interrupt/Next_is_Gap Reception Interrupt/Message Error Interrupt (IRR13): This interrupt assumes a different meaning depending on the RCAN-TL1 mode. It indicates that:

— The Timer (TCNTR) has overrun when RCAN-TL1 is working in event-trigger mode (including test modes)

- Time reference message with Next_is_Gap set has been received when working in time-trigger mode. Please note that when a Next_is_Gap is received the application is responsible to stop all transmission at the end of the current basic cycle (including test modes)
- Message error has occurred when in test mode. Note: If a Message Overload condition occurs when in Test Mode, then this bit will not be set.

Bit 13: IRR13	Description
0	Timer (TCNTR) has not overrun in event-trigger mode (including test modes) (Initial value) Time reference message with Next_is_Gap has not been received in time-trigger mode (including test modes) Message error has not occurred in test mode [Clearing condition] Writing 1
1	[Setting condition] Timer (TCNTR) has overrun and changed from H'FFFF to H'0000 in event-trigger mode (including test modes) Time reference message with Next_is_Gap has been received in time-trigger mode (including test modes) Message error has occurred in test mode

Bit 12 – Bus activity while in sleep mode (IRR12): IRR12 indicates that a CAN bus activity is present. While the RCAN-TL1 is in sleep mode and a dominant bit is detected on the CAN bus, this bit is set. This interrupt is cleared by writing a '1' to this bit position. Writing a '0' has no effect. If auto wakeup is not used and this interrupt is not requested it needs to be disabled by the related interrupt mask register. If auto wake up is not used and this interrupt is requested it should be cleared only after recovering from sleep mode. This is to avoid that a new falling edge of the reception line causes the interrupt to get set again.

Please note that the setting time of this interrupt is different from the clearing time of GSR4.

Bit 12: IRR12	Description
0	Bus idle state (Initial value) [Clearing condition] Writing 1
1	CAN bus activity detected in RCAN-TL1 sleep mode [Setting condition] Dominant bit level detection on the Rx line while in sleep mode

Bit 11 — Timer Compare Match Interrupt 2 (IRR11): Indicates that a Compare-Match condition occurred to the Timer Compare Match Register 2 (TCMR2). When the value set in the TCMR2 matches to Cycle Time (TCMR2 = CYCTR), this bit is set.

Bit 11: IRR11	Description
0	Timer Compare Match has not occurred to the TCMR2 (initial value) [Clearing condition] Writing 1
1	Timer Compare Match has occurred to the TCMR2 [Setting condition] TCMR2 matches to Cycle Time (TCMR2 = CYCTR)

Bit 10 — Start of new system matrix Interrupt (IRR10): Indicates that a new system matrix is starting.

When CCR = 0, this bit is set at the successful completion of reception/transmission of time reference message. Please note that when CMAX = 0 this interrupt is set at every basic cycle.

Bit 10: IRR10	Description
0	A new system matrix is not starting (initial value) [Clearing condition] Writing 1
1	Cycle counter reached zero. [Setting condition] Reception/transmission of time reference message is successfully completed when CMAX!= 3'b111 and CCR = 0

Bit 9 – Message Overrun/Overwrite Interrupt Flag (IRR9): Flag indicating that a message has been received but the existing message in the matching Mailbox has not been read as the corresponding RXPR or RFPR is already set to '1' and not yet cleared by the CPU. The received message is either abandoned (overrun) or overwritten dependant upon the NMC (New Message Control) bit. This bit is cleared when all bit in UMSR (Unread Message Status Register) are cleared (by writing '1') or by setting MBIMR (MailBox interrupt Mast Register) for all UMSR flag set. It is also cleared by writing a '1' to all the correspondent bit position in MBIMR. Writing to this bit position has no effect.

Bit 9: IRR9	Description
0	No pending notification of message overrun/overwrite [Clearing condition] Clearing of all bit in UMSR/setting MBIMR for all UMSR set (initial value)
1	A receive message has been discarded due to overrun condition or a message has been overwritten [Setting condition] Message is received while the corresponding RXPR and/or RFPR = 1 and MBIMR = 0

Bit 8 - Mailbox Empty Interrupt Flag (IRR8): This bit is set when one of the messages set for transmission has been successfully sent (corresponding TXACK flag is set) or has been successfully aborted (corresponding ABACK flag is set). In Event Triggered mode the related TXPR is also cleared and this mailbox is now ready to accept a new message data for the next transmission. In Time Trigger mode TXPR for the Mailboxes from 30 to 24 is not cleared after a successful transmission in order to keep transmitting at each programmed basic cycle. In effect, this bit is set by an OR'ed signal of the TXACK and ABACK bits not masked by the corresponding MBIMR flag. Therefore, this bit is automatically cleared when all the TXACK and ABACK bits are cleared. It is also cleared by writing a '1' to all the correspondent bit position in MBIMR. Writing to this bit position has no effect.

Bit 8: IRR8	Description
0	Messages set for transmission or transmission cancellation request NOT progressed. (Initial value) [Clearing Condition] All the TXACK and ABACK bits are cleared/setting MBIMR for all TXACK and ABACK set
1	Message has been transmitted or aborted, and new message can be stored (in TT mode Mailbox 24 to 30 can be programmed with a new message only in case of abortion) [Setting condition] When a TXACK or ABACK bit is set (if related MBIMR = 0).

Bit 7 - Overload Frame (IRR7): Flag indicating that the RCAN-TL1 has detected a condition that should initiate the transmission of an overload frame. Note that in the condition of transmission being prevented, such as listen only mode, an Overload Frame will NOT be transmitted, but IRR7 will still be set. IRR7 remains asserted until reset by writing a '1' to this bit position - writing a '0' has no effect.

Bit 7: IRR7	Description
0	[Clearing condition] Writing 1 (Initial value)
1	[Setting conditions] Overload condition detected

Bit 6 - Bus Off Interrupt Flag (IRR6): This bit is set when RCAN-TL1 enters the Bus-off state or when RCAN-TL1 leaves Bus-off and returns to Error-Active. The cause therefore is the existing condition TEC ≥ 256 at the node or the end of the Bus-off recovery sequence (128X11 consecutive recessive bits) or the transition from Bus Off to Halt (automatic or manual). This bit remains set even if the RCAN-TL1 node leaves the bus-off condition, and needs to be explicitly cleared by S/W. The S/W is expected to read the GSR0 to judge whether RCAN-TL1 is in the bus-off or error active status. It is cleared by writing a '1' to this bit position even if the node is still bus-off. Writing a '0' has no effect.

Bit 6: IRR6	Description
0	[Clearing condition] Writing 1 (Initial value)
1	Enter Bus off state caused by transmit error or Error Active state returning from Bus-off [Setting condition] When TEC ≥ 256 or End of Bus-off after 128 × 11 consecutive recessive bits or transition from Bus Off to Halt

Bit 5 - Error Passive Interrupt Flag (IRR5): Interrupt flag indicating the error passive state caused by the transmit or receive error counter or by Error Passive forced by test mode. This bit is reset by writing a '1' to this bit position, writing a '0' has no effect. If this bit is cleared the node may still be error passive. Please note that the SW needs to check GSR0 and GSR5 to judge whether RCAN-TL1 is in Error Passive or Bus Off status.

Bit 5: IRR5	Description
0	[Clearing condition] Writing 1 (Initial value)
1	Error passive state caused by transmit/receive error [Setting condition] When TEC ≥ 128 or REC ≥ 128 or Error Passive test mode is used

Bit 4 - Receive Error Counter Warning Interrupt Flag (IRR4): This bit becomes set if the receive error counter (REC) reaches a value greater than 95 when RCAN-TL1 is not in the Bus Off status. The interrupt is reset by writing a '1' to this bit position, writing '0' has no effect.

Bit 4: IRR4	Description
0	[Clearing condition] Writing 1 (Initial value)
1	Error warning state caused by receive error [Setting condition] When REC ≥ 96 and RCAN-TL1 is not in Bus Off

Bit 3 - Transmit Error Counter Warning Interrupt Flag (IRR3): This bit becomes set if the transmit error counter (TEC) reaches a value greater than 95. The interrupt is reset by writing a '1' to this bit position, writing '0' has no effect.

Bit 3: IRR3	Description
0	[Clearing condition] Writing 1 (Initial value)
1	Error warning state caused by transmit error [Setting condition] When TEC ≥ 96

Bit 2 - Remote Frame Receive Interrupt Flag (IRR2): Flag indicating that a remote frame has been received in a mailbox. This bit is set if at least one receive mailbox, with related MBIMR not set, contains a remote frame transmission request. This bit is automatically cleared when all bits in the Remote Frame Receive Pending Register (RFPR), are cleared. It is also cleared by writing a '1' to all the correspondent bit position in MBIMR. Writing to this bit has no effect.

Bit 2: IRR2	Description
0	[Clearing condition] Clearing of all bits in RFPR (Initial value)
1	At least one remote request is pending [Setting condition] When remote frame is received and the corresponding MBIMR = 0

Bit 1 – Data Frame Received Interrupt Flag (IRR1): IRR1 indicates that there are pending Data Frames received. If this bit is set at least one receive mailbox contains a pending message. This bit is cleared when all bits in the Data Frame Receive Pending Register (RXPR) are cleared, i.e. there is no pending message in any receiving mailbox. It is in effect a logical OR of the RXPR flags from each configured receive mailbox with related MBIMR not set. It is also cleared by writing a '1' to all the correspondent bit position in MBIMR. Writing to this bit has no effect.

Bit 1: IRR1	Description
0	[Clearing condition] Clearing of all bits in RXPR (Initial value)
1	Data frame received and stored in Mailbox [Setting condition] When data is received and the corresponding MBIMR = 0

Bit 0 – Reset/Halt/Sleep Interrupt Flag (IRR0): This flag can get set for three different reasons. It can indicate that:

1. Reset mode has been entered after a SW (MCR0) or HW reset
2. Halt mode has been entered after a Halt request (MCR1)
3. Sleep mode has been entered after a sleep request (MCR5) has been made while in Halt mode.

The GSR may be read after this bit is set to determine which state RCAN-TL1 is in.

Important: When a Sleep mode request needs to be made, the Halt mode must be used beforehand. Please refer to the MCR5 description and Figure 18.15 Halt Mode/Sleep Mode.

IRR0 is set by the transition from "0" to "1" of GSR3 or GSR4 or by transition from Halt mode to Sleep mode. So, IRR0 is not set if RCAN-TL1 enters Halt mode again right after exiting from Halt mode, without GSR4 being cleared. Similarly, IRR0 is not set by direct transition from Sleep mode to Halt Request. At the transition from Halt/Sleep mode to Transition/Reception, clearing GSR4 needs (one-bit time - TSEG2) to (one-bit time * 2 - TSEG2).

In the case of Reset mode, IRR0 is set, however, the interrupt to the CPU is not asserted since IMR0 is automatically set by initialization.

Bit 0: IRR0	Description
0	[Clearing condition] Writing 1
1	Transition to S/W reset mode or transition to halt mode or transition to sleep mode (Initial value) [Setting condition] When reset/halt/sleep transition is completed after a reset (MCR0 or HW) or Halt mode (MCR1) or Sleep mode (MCR5) is requested

(5) Message Buffer Error Status Register (MBESR)

This register is a status register that indicates the generation of a parity error in RAM (control 0, LAFM, and DATA). When a parity error occurs, the MBEF bit is set to 1.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	MBEF
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit 15 to 1: Reserved. The written value should always be '0' and the returned value is '0'.

Bit 0 – Message Buffer Parity Error Detection Status Flag (MBEF): Indicates the occurrence of a parity error with a data transfer. This bit is cleared by writing 1. Writing 0 to this bit is ignored.

Note: The MBESR register is not initialized when the value of the MCR0 bit in the master control register changes from 0 to 1.

Bit 0: MBEF	Description
0	Parity error does not occur
1	Parity error occurs

(6) Interrupt Mask Register (IMR)

The interrupt mask register is a 16 bit register that protects all corresponding interrupts in the Interrupt Request Register (IRR) from generating an output signal on the IRQ. An interrupt request is masked if the corresponding bit position is set to '1'. This register can be read or written at any time. The IMR directly controls the generation of IRQ, but does not prevent the setting of the corresponding bit in the IRR.

- IMR (Address = H'00A)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IMR15	IMR14	IMR13	IMR12	IMR11	IMR10	IMR9	IMR8	IMR7	IMR6	IMR5	IMR4	IMR3	IMR2	IMR1	IMR0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 15 to 0: Maskable interrupt sources corresponding to IRR[15:0] respectively. When a bit is set, the interrupt signal is not generated, although setting the corresponding IRR bit is still performed.

Bit[15:0]: IMRn	Description
0	Corresponding IRR is not masked (IRQ is generated for interrupt conditions)
1	Corresponding interrupt of IRR is masked (Initial value)

(7) Message Buffer Error Control Register (MBECR)

This register is a readable/writable register that protects an interrupt in message buffer error status register (MBESR). An interrupt request is masked when the MBIM bit is set to 1. The MBIM bit controls the interrupt request, but does not prevent the setting of the MBIM bit.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	MBIM
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit 15 to 1: Reserved. The written value should always be '0' and the returned value is '0'.

Bit 0 – Message Buffer Parity Error Detection Interrupt Mask (MBIM): When a bit is set, the interrupt signal is not generated, although setting the MBEF bit is still performed.

Note: The MBECR register is not initialized when the value of the MCR0 bit in the master control register changes from 0 to 1.

Bit 0: MBIM	Description
0	MBEF interrupt is not masked
1	MBEF interrupt is masked

(8) Transmit Error Counter (TEC) and Receive Error Counter (REC)

The Transmit Error Counter (TEC) and Receive Error Counter (REC) is a 16-bit read/(write) register that functions as a counter indicating the number of transmit/receive message errors on the CAN Interface. The count value is stipulated in the CAN protocol specification Refs. [1], [2], [3] and [4]. When not in (Write Error Counter) test mode this register is read only, and can only be modified by the CAN Interface. This register can be cleared by a Reset request (MCR0) or entering to bus off.

In Write Error Counter test mode (i.e. TST[2:0] = 3'b100), it is possible to write to this register. The same value can only be written to TEC/REC, and the value written into TEC is set to TEC and REC. When writing to this register, RCAN-TL1 needs to be put into Halt Mode. This feature is only intended for test purposes.

- TEC/REC (Address = H'00C)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0	REC7	REC6	REC5	REC4	REC3	REC2	REC1	REC0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Note: * It is only possible to write the value in test mode when TST[2:0] in MCR is 3'b100. REC is incremented during Bus Off to count the recurrences of 11 recessive bits as requested by the Bus Off recovery sequence.

18.3.4 RCAN-TL1 Mailbox Registers

The following sections describe RCAN-TL1 Mailbox registers that control/flag individual Mailboxes. The address is mapped as follows.

Important: LongWord access is carried out as two consecutive Word accesses.

32-Mailboxes version

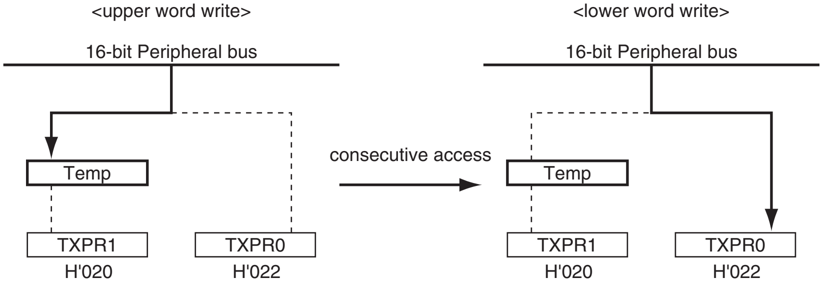
Description	Address	Name	Access Size (bits)
Transmit Pending 1	020	TXPR1	LW
Transmit Pending 0	022	TXPR0	—
	024		
	026		
Transmit Cancel 1	028	TXCR1	Word/LW
Transmit Cancel 0	02A	TXCR0	Word
	02C		
	02E		
Transmit Acknowledge 1	030	TXACK1	Word/LW
Transmit Acknowledge 0	032	TXACK0	Word
	034		
	036		
Abort Acknowledge 1	038	ABACK1	Word/LW
Abort Acknowledge 0	03A	ABACK0	Word
	03C		
	03E		
Data Frame Receive Pending 1	040	RXPR1	Word/LW
Data Frame Receive Pending 0	042	RXPR0	Word
	044		
	046		
Remote Frame Receive Pending 1	048	RFPR1	Word/LW
Remote Frame Receive Pending 0	04A	RFPR0	Word
	04C		
	04E		
Mailbox Interrupt Mask Register 1	050	MBIMR1	Word/LW
Mailbox Interrupt Mask Register 0	052	MBIMR0	Word
	054		
	056		
Unread message Status Register 1	058	UMSR1	Word/LW
Unread message Status Register 0	05A	UMSR0	Word
	05C		
	05E		

Figure 18.11 RCAN-TL1 Mailbox registers

(1) Transmit Pending Register (TXPR1, TXPR0)

The concatenation of TXPR1 and TXPR0 is a 32-bit register that contains any transmit pending flags for the CAN module. In the case of 16-bit bus interface, Long Word access is carried out as two consecutive word accesses.

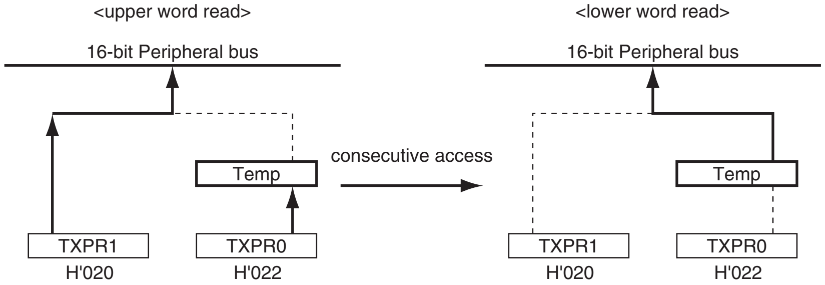
<Longword Write Operation>



Data is stored into Temp instead of TXPR1.

Longword data are stored into both TXPR1 and TXPR0 at the same time.

<Longword Read Operation>



TXPR0 is stored into Temp, when TXPR1 is read.

Temp is read instead of TXPR0.

The TXPR1 controls Mailbox-31 to Mailbox-16, and the TXPR0 controls Mailbox-15 to Mailbox-1. The CPU may set the TXPR bits to affect any message being considered for transmission by writing a '1' to the corresponding bit location. Writing a '0' has no effect, and TXPR cannot be cleared by writing a '0' and must be cleared by setting the corresponding TXCR bits. TXPR may be read by the CPU to determine which, if any, transmissions are pending or in progress. In effect there is a transmit pending bit for all Mailboxes except for the Mailbox-0. Writing a '1' to a bit location when the mailbox is not configured to transmit is not allowed.

In Event Triggered Mode RCAN-TL1 will clear a transmit pending flag after successful transmission of its corresponding message or when a transmission abort is requested successfully from the TXCR. In Time Trigger Mode, TXPR for the Mailboxes from 30 to 24 is NOT cleared after a successful transmission, in order to keep transmitting at each programmed basic cycle. The TXPR flag is not cleared if the message is not transmitted due to the CAN node losing the arbitration process or due to errors on the CAN bus, and RCAN-TL1 automatically tries to transmit it again unless its DART bit (Disable Automatic Re-Transmission) is set in the Message-Control of the corresponding Mailbox. In such case (DART set), the transmission is cleared and notified through Mailbox Empty Interrupt Flag (IRR8) and the correspondent bit within the Abort Acknowledgement Register (ABACK).

If the status of the TXPR changes, the RCAN-TL1 shall ensure that in the identifier priority scheme (MCR2 = 0), the highest priority message is always presented for transmission in an intelligent way even under circumstances such as bus arbitration losses or errors on the CAN bus. Please refer to the Application Note for details.

When the RCAN-TL1 changes the state of any TXPR bit position to a '0', an empty slot interrupt (IRR8) may be generated. This indicates that either a successful or an aborted mailbox transmission has just been made. If a message transmission is successful it is signalled in the TXACK register, and if a message transmission abortion is successful it is signalled in the ABACK register. By checking these registers, the contents of the Message of the corresponding Mailbox may be modified to prepare for the next transmission.

- TXPR1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXPR1[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Note: * It is possible only to write a '1' for a Mailbox configured as transmitter.

Bit 15 to 0 — Requests the corresponding Mailbox to transmit a CAN Frame. The bit 15 to 0 corresponds to Mailbox-31 to 16 respectively. When multiple bits are set, the order of the transmissions is governed by the MCR2 – CAN-ID or Mailbox number.

Bit[15:0]: TXPR1	Description
0	Transmit message idle state in corresponding mailbox (Initial value) [Clearing Condition] Completion of message transmission (for Event Triggered Messages) or message transmission abortion (automatically cleared)
1	Transmission request made for corresponding mailbox

- TXPR0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXPR0[15:1]															-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R

Note: * It is possible only to write a '1' for a Mailbox configured as transmitter.

Bit 15 to 1 — Indicates that the corresponding Mailbox is requested to transmit a CAN Frame. The bit 15 to 1 corresponds to Mailbox-15 to 1 respectively. When multiple bits are set, the order of the transmissions is governed by the MCR2 – CAN-ID or Mailbox number.

Bit[15:1]: TXPR0	Description
0	Transmit message idle state in corresponding mailbox (Initial value) [Clearing Condition] Completion of message transmission (for Event Triggered Messages) or message transmission abortion (automatically cleared)
1	Transmission request made for corresponding mailbox

Bit 0— Reserved: This bit is always '0' as this is a receive-only Mailbox. Writing a '1' to this bit position has no effect. The returned value is '0'.

(2) Transmit Cancel Register (TXCR1, TXCR0)

The TXCR1 and TXCR0 are 16-bit read/conditionally-write registers. The TXCR1 controls Mailbox-31 to Mailbox-16, and the TXCR0 controls Mailbox-15 to Mailbox-1.This register is used by the CPU to request the pending transmission requests in the TXPR to be cancelled. To clear the corresponding bit in the TXPR the CPU must write a '1' to the bit position in the TXCR. Writing a '0' has no effect.

When an abort has succeeded the CAN controller clears the corresponding TXPR + TXCR bits, and sets the corresponding ABACK bit. However, once a Mailbox has started a transmission, it cannot be cancelled by this bit. In such a case, if the transmission finishes in success, the CAN controller clears the corresponding TXPR + TXCR bit, and sets the corresponding TXACK bit, however, if the transmission fails due to a bus arbitration loss or an error on the bus, the CAN controller clears the corresponding TXPR + TXCR bit, and sets the corresponding ABACK bit. If an attempt is made by the CPU to clear a mailbox transmission that is not transmit-pending it has no effect. In this case the CPU will be not able at all to set the TXCR flag.

- TXCR1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXCR1[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Note: * Only writing a '1' to a Mailbox that is requested for transmission and is configured as transmit.

Bit 15 to 0 — Requests the corresponding Mailbox, that is in the queue for transmission, to cancel its transmission. The bit 15 to 0 corresponds to Mailbox-31 to 16 (and TXPR1[15:0]) respectively.

Bit[15:0]:TXCR1	Description
0	Transmit message cancellation idle state in corresponding mailbox (Initial value) [Clearing Condition] Completion of transmit message cancellation (automatically cleared)
1	Transmission cancellation request made for corresponding mailbox

- TXCR0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXCR0[15:1]															-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R

Note: * Only writing a '1' to a Mailbox that is requested for transmission and is configured as transmit.

Bit 15 to 1 — Requests the corresponding Mailbox, that is in the queue for transmission, to cancel its transmission. The bit 15 to 1 corresponds to Mailbox-15 to 1 (and TXPR0[15:1]) respectively.

Bit[15:1]: TXCR0	Description
0	Transmit message cancellation idle state in corresponding mailbox (Initial value) [Clearing Condition] Completion of transmit message cancellation (automatically cleared)
1	Transmission cancellation request made for corresponding mailbox

Bit 0 — This bit is always '0' as this is a receive-only mailbox. Writing a '1' to this bit position has no effect and always read back as a '0'.

(3) Transmit Acknowledge Register (TXACK1, TXACK0)

The TXACK1 and TXACK0 are 16-bit read/conditionally-write registers. These registers are used to signal to the CPU that a mailbox transmission has been successfully made. When a transmission has succeeded the RCAN-TL1 sets the corresponding bit in the TXACK register. The CPU may clear a TXACK bit by writing a '1' to the corresponding bit location. Writing a '0' has no effect.

- TXACK1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXACK1[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Note: * Only when writing a '1' to clear.

Bit 15 to 0 — Notifies that the requested transmission of the corresponding Mailbox has been finished successfully. The bit 15 to 0 corresponds to Mailbox-31 to 16 respectively.

Bit[15:0]:TXACK1	Description
0	[Clearing Condition] Writing '1' (Initial value)
1	Corresponding Mailbox has successfully transmitted message (Data or Remote Frame) [Setting Condition] Completion of message transmission for corresponding mailbox

- TXACK0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXACK0[15:1]															-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	-

Note: * Only when writing a '1' to clear.

Bit 15 to 1 — Notifies that the requested transmission of the corresponding Mailbox has been finished successfully. The bit 15 to 1 corresponds to Mailbox-15 to 1 respectively.

Bit[15:1]:TXACK0	Description
0	[Clearing Condition] Writing '1' (Initial value)
1	Corresponding Mailbox has successfully transmitted message (Data or Remote Frame) [Setting Condition] Completion of message transmission for corresponding mailbox

Bit 0 — This bit is always '0' as this is a receive-only mailbox. Writing a '1' to this bit position has no effect and always read back as a '0'.

(4) Abort Acknowledge Register (ABACK1, ABACK0)

The ABACK1 and ABACK0 are 16-bit read/conditionally-write registers. These registers are used to signal to the CPU that a mailbox transmission has been aborted as per its request. When an abort has succeeded the RCAN-TL1 sets the corresponding bit in the ABACK register. The CPU may clear the Abort Acknowledge bit by writing a '1' to the corresponding bit location. Writing a '0' has no effect. An ABACK bit position is set by the RCAN-TL1 to acknowledge that a TXPR bit has been cleared by the corresponding TXCR bit.

- ABACK1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ABACK1[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Note: * Only when writing a '1' to clear.

Bit 15 to 0 — Notifies that the requested transmission cancellation of the corresponding Mailbox has been performed successfully. The bit 15 to 0 corresponds to Mailbox-31 to 16 respectively.

Bit[15:0]:ABACK1	Description
0	[Clearing Condition] Writing '1' (Initial value)
1	Corresponding Mailbox has cancelled transmission of message (Data or Remote Frame) [Setting Condition] Completion of transmission cancellation for corresponding mailbox

- ABACK0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ABACK0[15:1]															-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R

Note: * Only when writing a '1' to clear.

Bit 15 to 1 — Notifies that the requested transmission cancellation of the corresponding Mailbox has been performed successfully. The bit 15 to 1 corresponds to Mailbox-15 to 1 respectively.

Bit[15:1]:ABACK0	Description
0	[Clearing Condition] Writing '1' (Initial value)
1	Corresponding Mailbox has cancelled transmission of message (Data or Remote Frame) [Setting Condition] Completion of transmission cancellation for corresponding mailbox

Bit 0 — This bit is always '0' as this is a receive-only mailbox. Writing a '1' to this bit position has no effect and always read back as a '0'.

(5) Data Frame Receive Pending Register (RXPR1, RXPR0)

The RXPR1 and RXPR0 are 16-bit read/conditionally-write registers. The RXPR is a register that contains the received Data Frames pending flags associated with the configured Receive Mailboxes. When a CAN Data Frame is successfully stored in a receive mailbox the corresponding bit is set in the RXPR. The bit may be cleared by writing a '1' to the corresponding bit position. Writing a '0' has no effect. However, the bit may only be set if the mailbox is configured by its MBC (Mailbox Configuration) to receive Data Frames. When a RXPR bit is set, it also sets IRR1 (Data Frame Received Interrupt Flag) if its MBIMR (Mailbox Interrupt Mask Register) is not set, and the interrupt signal is generated if IMR1 is not set. Please note that these bits are only set by receiving Data Frames and not by receiving Remote frames.

- RXPR1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RXPR1[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Note : * Only when writing a '1' to clear.

Bit 15 to 0 — Configurable receive mailbox locations corresponding to each mailbox position from 31 to 16 respectively.

Bit[15:0]: RXPR1	Description
0	[Clearing Condition] Writing '1' (Initial value)
1	Corresponding Mailbox received a CAN Data Frame [Setting Condition] Completion of Data Frame receive on corresponding mailbox

- RXPR0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RXPR0[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Note: * Only when writing a '1' to clear.

Bit 15 to 0 — Configurable receive mailbox locations corresponding to each mailbox position from 15 to 0 respectively.

Bit[15:0]: RXPR0	Description
0	[Clearing Condition] Writing '1' (Initial value)
1	Corresponding Mailbox received a CAN Data Frame [Setting Condition] Completion of Data Frame receive on corresponding mailbox

(6) Remote Frame Receive Pending Register (RFPR1, RFPR0)

The RFPR1 and RFPR0 are 16-bit read/conditionally-write registers. The RFPR is a register that contains the received Remote Frame pending flags associated with the configured Receive Mailboxes. When a CAN Remote Frame is successfully stored in a receive mailbox the corresponding bit is set in the RFPR. The bit may be cleared by writing a '1' to the corresponding bit position. Writing a '0' has no effect. In effect there is a bit position for all mailboxes. However, the bit may only be set if the mailbox is configured by its MBC (Mailbox Configuration) to receive Remote Frames. When a RFPR bit is set, it also sets IRR2 (Remote Frame Receive Interrupt Flag) if its MBIMR (Mailbox Interrupt Mask Register) is not set, and the interrupt signal is generated if IMR2 is not set. Please note that these bits are only set by receiving Remote Frames and not by receiving Data frames.

- RFPR1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFPR1[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Note: * Only when writing a '1' to clear.

Bit 15 to 0 — Remote Request pending flags for mailboxes 31 to 16 respectively.

Bit[15:0]: RFPR1	Description
0	[Clearing Condition] Writing '1' (Initial value)
1	Corresponding Mailbox received Remote Frame [Setting Condition] Completion of remote frame receive in corresponding mailbox

- RFPR0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFPR0[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Note: * Only when writing a '1' to clear.

Bit 15 to 0 — Remote Request pending flags for mailboxes 15 to 0 respectively.

Bit[15:0]: RFPR0	Description
0	[Clearing Condition] Writing '1' (Initial value)
1	Corresponding Mailbox received Remote Frame [Setting Condition] Completion of remote frame receive in corresponding mailbox

(7) Mailbox Interrupt Mask Register (MBIMR)

The MBIMR1 and MBIMR0 are 16-bit read/write registers. The MBIMR only prevents the setting of IRR related to the Mailbox activities, that are IRR[1] – Data Frame Received Interrupt, IRR[2] – Remote Frame Receive Interrupt, IRR[8] – Mailbox Empty Interrupt, and IRR[9] – Message OverRun/OverWrite Interrupt. If a mailbox is configured as receive, a mask at the corresponding bit position prevents the generation of a receive interrupt (IRR[1] and IRR[2] and IRR[9]) but does not prevent the setting of the corresponding bit in the RXPR or RFPR or UMSR. Similarly when a mailbox has been configured for transmission, a mask prevents the generation of an Interrupt signal and setting of an Mailbox Empty Interrupt due to successful transmission or abortion of transmission (IRR[8]), however, it does not prevent the RCAN-TL1 from clearing the corresponding TXPR/TXCR bit + setting the TXACK bit for successful transmission, and it does not prevent the RCAN-TL1 from clearing the corresponding TXPR/TXCR bit + setting the ABACK bit for abortion of the transmission.

A mask is set by writing a '1' to the corresponding bit position for the mailbox activity to be masked. At reset all mailbox interrupts are masked.

- MBIMR1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBIMR1[15:0]															
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 15 to 0 — Enable or disable interrupt requests from individual Mailbox-31 to Mailbox-16 respectively.

Bit[15:0]: MBIMR1	Description
0	Interrupt Request from IRR1/IRR2/IRR8/IRR9 enabled
1	Interrupt Request from IRR1/IRR2/IRR8/IRR9 disabled (initial value)

- MBIMR0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBIMR0[15:0]															
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 15 to 0 — Enable or disable interrupt requests from individual Mailbox-15 to Mailbox-0 respectively.

Bit[15:0]: MBIMR0	Description
0	Interrupt Request from IRR1/IRR2/IRR8/IRR9 enabled
1	Interrupt Request from IRR1/IRR2/IRR8/IRR9 disabled (initial value)

(8) Unread Message Status Register (UMSR)

This register is a 32-bit read/conditionally write register and it records the mailboxes whose contents have not been accessed by the CPU prior to a new message being received. If the CPU has not cleared the corresponding bit in the RXPR or RFPR when a new message for that mailbox is received, the corresponding UMSR bit is set to '1'. This bit may be cleared by writing a '1' to the corresponding bit location in the UMSR. Writing a '0' has no effect.

If a mailbox is configured as transmit box, the corresponding UMSR will not be set.

- UMSR1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UMSR1[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Note: * Only when writing a '1' to clear.

Bit 15 to 0 — Indicate that an unread received message has been overwritten or overrun condition has occurred for Mailboxes 31 to 16.

Bit[15:0]: UMSR1	Description
0	[Clearing Condition] Writing '1' (initial value)
1	Unread received message is overwritten by a new message or overrun condition [Setting Condition] When a new message is received before RXPR or RFPR is cleared

- UMSR0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UMSR0[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Note: * Only when writing a '1' to clear.

Bit 15 to 0 — Indicate that an unread received message has been overwritten or overrun condition has occurred for Mailboxes 15 to 0.

Bit[15:0]: UMSR0	Description
0	[Clearing Condition] Writing '1' (initial value)
1	Unread received message is overwritten by a new message or overrun condition [Setting Condition] When a new message is received before RXPR or RFPR is cleared

18.3.5 Timer Registers

The Timer is 16 bits and supports several source clocks. A pre-scale counter can be used to reduce the speed of the clock. It also supports three Compare Match Registers (TCMR2, TCMR1, TCMR0). The address map is as follows.

Important: These registers can only be accessed in Word size (16-bit).

Description	Address	Name	Access Size (bits)
TimerTrigger Control Register 0	080	TTCR0	Word (16)
Cycle Maximum/Tx-Enable Window Register	084	CMAX_TEW	Word (16)
Reference Trigger Offset Register	086	RFTROFF	Word (16)
Timer Status Register	088	TSR	Word (16)
Cycle Counter Register	08A	CCR	Word (16)
Timer Counter Register	08C	TCNTR	Word (16)
Cycle Time Register	090	CYCTR	Word (16)
Reference Mark Register	094	RFMK	Word (16)
Timer Compare Match Register 0	098	TCMR0	Word (16)
Timer Compare Match Register 1	09C	TCMR1	Word (16)
Timer Compare Match Register 2	0A0	TCMR2	Word (16)
Tx-Trigger Time Selection Register	0A4	TTTSEL	Word (16)

Figure 18.12 RCAN-TL1 Timer registers

(1) Time Trigger Control Register0 (TTCR0)

The Time Trigger Control Register0 is a 16-bit read/write register and provides functions to control the operation of the Timer. When operating in Time Trigger Mode, please refer to section 18.4.3 (1), Time Triggered Transmission.

- TTCR0 (Address = H'080)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TCR15	TCR14	TCR13	TCR12	TCR11	TCR10	-	-	-	TCR6	TPSC5	TPSC4	TPSC3	TPSC2	TPSC1	TPSC0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 15 — Enable Timer: When this bit is set, the timer TCNTR is running. When this bit is cleared, TCNTR and CCR are cleared.

Bit15: TTCR0 15	Description
0	Timer and CCR are cleared and disabled (initial value)
1	Timer is running

Bit 14 — TimeStamp value: Specifies if the Timestamp for transmission and reception in Mailboxes 15 to 1 must contain the Cycle Time (CYCTR) or the concatenation of CCR[5:0] + CYCTR[15:6]. This feature is very useful for time triggered transmission to monitor Rx_Trigger.

This register does not affect the TimeStamp for Mailboxes 30 and 31.

Bit14: TTCR0 14	Description
0	CYCTR[15:0] is used for the TimeStamp in Mailboxes 15 to 1 (initial value)
1	CCR[5:0] + CYCTR[15:6] is used for the TimeStamp in Mailboxes 15 to 1

Bit 13 — Cancellation by TCMR2: The messages in the transmission queue are cancelled by setting TXCR, when both this bit and bit12 are set and compare match occurs when RCAN-TL1 is not in the Halt status, causing the setting of all TXCR bits with the corresponding TXPR bits set.

Bit13: TTCR0 13	Description
0	Cancellation by TCMR2 compare match is disabled (initial value)
1	Cancellation by TCMR2 compare match is enabled

Bit 12 — TCMR2 compare match enable: When this bit is set, IRR11 is set by TCMR2 compare match.

Bit12 TTCR0 12	Description
0	IRR11 isn't set by TCMR2 compare match (initial value)
1	IRR11 is set by TCMR2 compare match

Bit 11 — TCMR1 compare match enable: When this bit is set, IRR15 is set by TCMR1 compare match.

Bit11 TTCR0 11	Description
0	IRR15 isn't set by TCMR1 compare match (initial value)
1	IRR15 is set by TCMR1 compare match

Bit 10 — TCMR0 compare match enable: When this bit is set, IRR14 is set by TCMR0 compare match.

Bit10 TTCR0 10	Description
0	IRR14 isn't set by TCMR0 compare match (initial value)
1	IRR14 is set by TCMR0 compare match

Bits 9 to 7: Reserved. The written value should always be '0' and the returned value is '0'.

Bit 6 — Timer Clear-Set Control by TCMR0: Specifies if the Timer is to be cleared and set to H'0000 when the TCMR0 matches to the TCNTR. Please note that the TCMR0 is also capable to generate an interrupt signal to the CPU via IRR14.

Note: If RCAN-TL1 is working in TTCAN mode (CMAX isn't 3'b111), TTCR0 bit6 has to be '0' to avoid clearing Local Time.

Bit6: TTCR0 6	Description
0	Timer is not cleared by the TCMR0 (initial value)
1	Timer is cleared by the TCMR0

Bit5 to 0 — RCAN-TL1 Timer Prescaler (TPSC[5:0]): This control field allows the timer source clock (4*[RCAN-TL1 system clock]) to be divided before it is used for the timer. This function is available only in event-trigger mode. In time trigger mode (CMAX is not 3'b111), one nominal Bit Timing (= one bit length of CAN bus) is automatically chosen as source clock of TCNTR.

The following relationship exists between source clock period and the timer period.

Bit[5:0]: TPSC[5:0]	Description
0 0 0 0 0 0	1 X Source Clock (initial value)
0 0 0 0 0 1	2 X Source Clock
0 0 0 0 1 0	3 X Source Clock
0 0 0 0 1 1	4 X Source Clock
0 0 0 1 0 0	5 X Source Clock
. . . . . .	. . . . . .
. . . . . .	. . . . . .
1 1 1 1 1 1	64 X Source Clock

(2) Cycle Maximum/Tx-Enable Window Register (CMAX_TEW)

This register is a 16-bit read/write register. CMAX specifies the maximum value for the cycle counter (CCR) for TT Transmissions to set the number of basic cycles in the matrix system. When the Cycle Counter reaches the maximum value (CCR = CMAX), after a full basic cycle, it is cleared to zero and an interrupt is generated on IRR.10.

TEW specifies the width of Tx-Enable window.

- CMAX_TEW (Address = H'084)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	CMAX[2:0]			-	-	-	-	TEW[3:0]			
Initial value:	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Bits 15 to 11: Reserved. The written value should always be '0' and the returned value is '0'.

Bit 10 to 8 — Cycle Count Maximum (CMAX): Indicates the maximum number of CCR. The number of basic cycles available in the matrix cycle for Timer Triggered transmission is (Cycle Count Maximum + 1).

Unless CMAX = 3'b111, RCAN-TL1 is in time-trigger mode and time trigger function is available. If CMAX = 3'b111, RCAN-TL1 is in event-trigger mode.

Bit[10:8]: CMAX[2:0]	Description
0 0 0	Cycle Count Maximum = 0
0 0 1	Cycle Count Maximum = 1
0 1 0	Cycle Count Maximum = 3
0 1 1	Cycle Count Maximum = 7
1 0 0	Cycle Count Maximum = 15
1 0 1	Cycle Count Maximum = 31
1 1 0	Cycle Count Maximum = 63
1 1 1	CCR is cleared and RCAN-TL1 is in event-trigger mode. (initial value)

Important: Please set CMAX = 3'b111 when event-trigger mode is used.

Bits 7 to 4: Reserved. The written value should always be '0' and the returned value is '0'.

Bit 3 to 0 — Tx-Enable Window (TEW): Indicates the width of Tx-Enable Window. TEW = H'00 shows the width is one nominal Bit Timing. All values from 0 to 15 are allowed to be set.

Bit[3:0]: TEW[3:0]	Description
0 0 0 0	The width of Tx-Enable Window = 1 (initial value)
0 0 0 1	The width of Tx-Enable Window = 2
0 0 1 0	The width of Tx-Enable Window = 3
0 0 1 1	The width of Tx-Enable Window = 4
. . . .	. . . . . .
. . . .	. . . . . .
1 1 1 1	The width of Tx-Enable Window = 16

Note: The CAN core always needs a time between 1 to 2 bit timing to initiate transmission. The above values are not considering this accuracy.

(3) Reference Trigger Offset Register (RFTROFF)

This is a 8-bit read/write register that affects Tx-Trigger Time (TTT) of Mailbox-30. The TTT of Mailbox-30 is compared with CYCTR after RFTROFF extended with sign is added to the TTT. However, the value of TTT is not modified. The offset value doesn't affect others except Mailbox-30.

- RFTROFF (Address = H'086)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFTROFF[7:0]								-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Bit 15 to 8 — Indicate the value of Reference Trigger Offset.

Bits 7 to 0: Reserved. The written value should always be '0' and the returned value is '0'.

Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Description
0	0	0	0	0	0	0	0	Ref_trigger_offset = +0 (initial value)
0	0	0	0	0	0	0	1	Ref_trigger_offset = +1
0	0	0	0	0	0	1	0	Ref_trigger_offset = +2
.	.	.	.	.	.	.	.	
0	1	1	1	1	1	1	1	Ref_trigger_offset = +127
.	.	.	.	.	.	.	.	
1	1	1	1	1	1	1	1	Ref_trigger_offset = –1
1	1	1	1	1	1	1	0	Ref_trigger_offset = –2
.	.	.	.	.	.	.	.	
1	0	0	0	0	0	0	1	Ref_trigger_offset = –127
1	0	0	0	0	0	0	0	Prohibited

(4) Timer Status Register (TSR)

This register is a 16-bit read-only register, and allows the CPU to monitor the Timer Compare Match status and the Timer Overrun Status.

- TSR (Address = H'088)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	TSR4	TSR3	TSR2	TSR1	TSR0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bits 15 to 5: Reserved. The written value should always be '0' and the returned value is '0'.

Bit 4 to 0 — RCAN-TL1 Timer Status (TSR[4:0]): This read-only field allows the CPU to monitor the status of the Cycle Counter, the Timer and the Compare Match registers. Writing to this field has no effect.

Bit 4 — Start of New System Matrix (TSR4): Indicates that a new system matrix is starting. When CCR = 0, this bit is set at the successful completion of reception/transmission of time reference message.

Bit4: TSR4	Description
0	A new system matrix is not starting (initial value) [Clearing condition] Writing '1' to IRR10 (Cycle Counter Overflow Interrupt)
1	Cycle counter reached zero [Setting condition] When the Cycle Counter value changes from the maximum value (CMAX) to H'0. Reception/transmission of time reference message is successfully completed when CMAX!= 3'b111 and CCR = 0

Bit 3 — Timer Compare Match Flag 2 (TSR3): Indicates that a Compare-Match condition occurred to the Timer Compare Match Register 2 (TCMR2). When the value set in the TCMR2 matches to Cycle Time Register (TCMR2 = CYCTR), this bit is set if TTCR0 bit12 = 1. Please note that this bit is read-only and is cleared when IRR11 (Timer Compare Match Interrupt 2) is cleared.

Bit3: TSR3	Description
0	Timer Compare Match has not occurred to the TCMR2 (Initial value) [Clearing condition] Writing '1' to IRR11 (Timer Compare Match Interrupt 1)
1	Timer Compare Match has occurred to the TCMR2 [Setting condition] TCMR2 matches to Cycle Time (TCMR2 = CYCTR), if TTCR0 bit12 = 1.

Bit 2 — Timer Compare Match Flag 1 (TSR2): Indicates that a Compare-Match condition occurred to the Timer Compare Match Register 1 (TCMR1). When the value set in the TCMR1 matches to Cycle Time Register (TCMR1 = CYCTR), this bit is set if TTCR0 bit11 = 1. Please note that this bit is read-only and is cleared when IRR15 (Timer Compare Match Interrupt 1) is cleared.

Bit2: TSR2	Description
0	Timer Compare Match has not occurred to the TCMR1 (Initial value) [Clearing condition] Writing '1' to IRR15 (Timer Compare Match Interrupt 1)
1	Timer Compare Match has occurred to the TCMR1 [Setting condition] TCMR1 matches to Cycle Time (TCMR1 = CYCTR), if TTCR0 bit11 = 1.

Bit 1 — Timer Compare Match Flag 0 (TSR1): Indicates that a Compare-Match condition occurred to the Compare Match Register 0 (TCMR0). When the value set in the TCMR0 matches to the Timer value (TCMR0 = TCNTR), this bit is set if TTCR0 bit10 = 1. Please note that this bit is read-only and is cleared when IRR14 (Timer Compare Match Interrupt 0) is cleared.

Bit1: TSR1	Description
0	Compare Match has not occurred to the TCMR0 (Initial value) [Clearing condition] Writing '1' to IRR14 (Timer Compare Match Interrupt 0)
1	Compare Match has occurred to the TCMR0 [Setting condition] TCMR0 matches to the Timer value (TCMR0 = TCNTR)

Bit 0 — Timer Overrun/Next_is_Gap Reception/Message Error (TSR0): This flag is assigned to three different functions. It indicates that the Timer has overrun when working in event-trigger mode, time reference message with Next_is_Gap set has been received in time-trigger mode, and error detected on the CAN bus has occurred in test mode, respectively. Test mode has higher priority with respect to the other settings.

Bit0: TSR0	Description
0	Timer (TCNTR) has not overrun in event-trigger mode (Initial value) Time reference message with Next_is_Gap has not been received in time-trigger mode message error has not occurred in test mode. [Clearing condition] Writing '1' to IRR13
1	[Setting condition] Timer (TCNTR) has overrun and changed from H'FFFF to H'0000 in event-trigger mode.time reference message with Next_is_Gap has been received in time-trigger mode message error has occurred in test mode

(5) Cycle Counter Register (CCR)

This register is a 6-bit read/write register. Its purpose is to store the number of the basic cycle for Time -Triggered Transmissions. Its value is updated in different fashions depending if RCAN-TL1 is programmed to work as a potential time master or as a time slave. If RCAN-TL1 is working as (potential) time master, CCR is:

- Incremented by one every time the cycle time (CYCTR) matches to Tx-Trigger Time of Mailbox-30 or
- Overwritten with the value contained in MSG_DATA_0[5:0] of Mailbox 31 when a valid reference message is received.

If RCAN-TL1 is working as a time slave, CCR is only overwritten with the value of MSG_DATA_0[5:0] of Mailbox 31 when a valid reference message is received.

If CMAX = 3'111, CCR is always H'0000.

- CCR (Address = H'08A)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	CCR[5:0]					
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 to 6: Reserved. The written value should always be '0' and the returned value is '0'.

Bit 5 to 0 — Cycle Counter Register (CCR): Indicates the number of the current Base Cycle of the matrix cycle for Timer Triggered transmission.

(6) Timer Counter Register (TCNTR)

This is a 16-bit read/write register that allows the CPU to monitor and modify the value of the Free Running Timer Counter. When the Timer meets TCMR0 (Timer Compare Match Register 0) + TTCR0 [6] is set to '1', the TCNTR is cleared to H'0000 and starts running again. In Time-Trigger mode, this timer can be used as Local Time and TTCR0[6] has to be cleared to work as a free running timer.

Notes: 1. It is possible to write into this register only when it is enabled by the bit 15 in TTCR0. If TTCR0 bit15 = 0, TCNTR is always H'0000.

2. There could be a delay of a few clock cycles between the enabling of the timer and the moment where TCNTR starts incrementing. This is caused by the internal logic used for the pre-scaler.

- TCNTR (Address = H'08C)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TCNTR[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Note: * The register can be written only when enabled in TTCR0[15]. Write operation is not allowed in Time Trigger mode (i.e. CMAX is not 3'b111).

Bit 15 to 0 — Indicate the value of the Free Running Timer.

(7) Cycle Time register (CYCTR)

This register is a 16-bit read-only register. This register shows Cycle Time = Local Time (TCNTR) - Reference_Mark (RFMK). In ET mode this register is the exact copy of TCNTR as RFMK is always fixed to zero.

- CYCTR (Address = H'090)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CYCTR[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

(8) Reference Mark Register (RFMK)

This register is a 16-bit read-only register. The purpose of this register is to capture Local Time (TCNTR) at SOF of the reference message when the message is received or transmitted successfully. In ET mode this register is not used and it is always cleared to zero.

- RFMK (Address = H'094)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFMK[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit 15 to 0 — Reference Mark Register (RFMK): Indicates the value of TCNTR at SOF of time reference message.

(9) Timer Compare Match Registers (TCMR0, TCMR1, TCMR2)

These three registers are 16-bit read/write registers and are capable of generating interrupt signals, clearing-setting the Timer value (only supported by TCMR0) or clear the transmission messages in the queue (only supported by TCMR2). TCMR0 is compared with TCNTR, however, TCMR1 and TCMR2 are compared with CYCTR.

The value used for the compare can be configured independently for each register. In order to set flags, TTCR0 bit 12-10 needs to be set.

In Time-Trigger mode, TTCR0 bit6 has to be cleared by software to prevent TCNTR from being cleared.

TMCR0 is for Init_Watch_Trigger, and TCMR2 is for Watch_Trigger.

Interrupt:

The interrupts are flagged by the Bit11, Bit15 and 14 in the IRR accordingly when a Compare Match occurs, and setting these bits can be enabled by Bit12, Bit11, Bit10 in TTCR0. The generation of interrupt signals itself can be prevented by the Bit11, Bit15 and Bit14 in the IMR. When a Compare Match occurs and the IRR11 (or IRR15 or IRR14) is set, the Bit3 or Bit2 or Bit1 in the TSR (Timer Status Register) is also set. Clearing the IRR bit also clears the corresponding bit of TSR.

Timer Clear-Set:

The Timer value can only be cleared when a Compare Match occurs if it is enabled by the Bit6 in the TTCR0. TCMR1 and TCMR2 do not have this function.

Cancellation of the messages in the transmission queue:

The messages in the transmission queue can only be cleared by the TCMR2 through setting TXCR when a Compare Match occurs while RCAN-TL1 is not in the halt status. TCMR1 and TCMR0 do not have this function.

- TCMR0 (Address = H'098)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TCMR0[15:0]															
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 15 to 0 — Timer Compare Match Register (TCMR0): Indicates the value of TCNTR when compare match occurs.

- TCMR1 (Address = H'09C)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TCMR1[15:0]															
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 15 to 0 — Timer Compare Match Register (TCMR1): Indicates the value of CYCTR when compare match occurs.

- TCMR2 (Address = H'0A0)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TCMR2[15:0]															
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 15 to 0 — Timer Compare Match Register (TCMR2): Indicates the value of CYCTR when compare match occurs.

(10) Tx-Trigger Time Selection Register (TTTSEL)

This register is a 16-bit read/write register and specifies the Tx-Trigger Time waiting for compare match with Cycle Time. Only one bit is allowed to be set. Please don't set more bits than one, or clear all bits.

This register may only be modified during configuration mode. The modification algorithm is shown in Figure 18.13.

Please note that this register is only indented for test and diagnosis. When not in test mode, this register must not be written to and the returned value is not guaranteed.

- TTTSEL (Address = H'0A4)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	TTTSEL[14:8]							-	-	-	-	-	-	-	-
Initial value:	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Note: Only one bit is allowed to be set.

Bit 15: Reserved. The written value should always be '0' and the returned value is '0'.

Bit 14 to 8 — Specifies the Tx-Trigger Time waiting for compare match with CYCTR The bit 14 to 8 corresponds to Mailbox-30 to 24, respectively.

Bits 7 to 0: Reserved. The written value should always be '0' and the returned value is '0'.

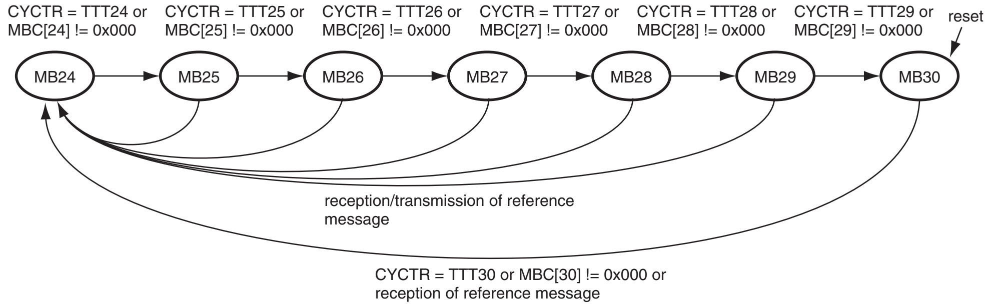


Figure 18.13 TTTSEL modification algorithm

18.4 Application Note

18.4.1 Test Mode Settings

The RCAN-TL1 has various test modes. The register TST[2:0] (MCR[10:8]) is used to select the RCAN-TL1 test mode. The default (initialized) settings allow RCAN-TL1 to operate in Normal mode. The following table is examples for test modes.

Test Mode can be selected only while in configuration mode. The user must then exit the configuration mode (ensuring BCR0/BCR1 is set) in order to run the selected test mode.

Bit10: TST2	Bit9: TST1	Bit8: TST0	Description
0	0	0	Normal Mode (initial value)
0	0	1	Listen-Only Mode (Receive-Only Mode)
0	1	0	Self Test Mode 1 (External)
0	1	1	Self Test Mode 2 (Internal)
1	0	0	Write Error Counter
1	0	1	Error Passive Mode
1	1	0	Setting prohibited
1	1	1	Setting prohibited

Normal Mode: RCAN-TL1 operates in the normal mode.

Listen-Only Mode: ISO-11898 requires this mode for baud rate detection. The Error Counters are cleared and disabled so that the TEC/REC does not increase the values, and the CTxn (n = A, B, C, and D) Output is disabled so that RCAN-TL1 does not generate error frames or acknowledgment bits. IRR13 is set when a message error occurs.

Self Test Mode 1: RCAN-TL1 generates its own Acknowledge bit, and can store its own messages into a reception mailbox (if required). The CRxn/CTxn (n = A, B, C, and D) pins must be connected to the CAN bus.

Self Test Mode 2: RCAN-TL1 generates its own Acknowledge bit, and can store its own messages into a reception mailbox (if required). The CRxn/CTxn (n = A, B, C, and D) pins do not need to be connected to the CAN bus or any external devices, as the internal CTxn (n = A, B, C, and D) is looped back to the internal CRxn (n = A, B, C, and D). CTxn (n = A, B, C, and D) pin outputs only recessive bits and CRxn (n = A, B, C, and D) pin is disabled.

Write Error Counter: TEC/REC can be written in this mode. RCAN-TL1 can be forced to become an Error Passive mode by writing a value greater than 127 into the Error Counters. The value written into TEC is used to write into REC, so only the same value can be set to these registers. Similarly, RCAN-TL1 can be forced to become an Error Warning by writing a value greater than 95 into them.

RCAN-TL1 needs to be in Halt Mode when writing into TEC/REC (MCR1 must be "1" when writing to the Error Counter). Furthermore this test mode needs to be exited prior to leaving Halt mode.

Error Passive Mode: RCAN-TL1 can be forced to enter Error Passive mode.
Note: The REC will not be modified by implementing this Mode. However, once running in Error Passive Mode, the REC will increase normally should errors be received. In this Mode, RCAN-TL1 will enter BusOff if TEC reaches 256 (Dec). However when this mode is used RCAN-TL1 will not be able to become Error Active. Consequently, at the end of the Bus Off recovery sequence, RCAN-TL1 will move to Error Passive and not to Error Active.

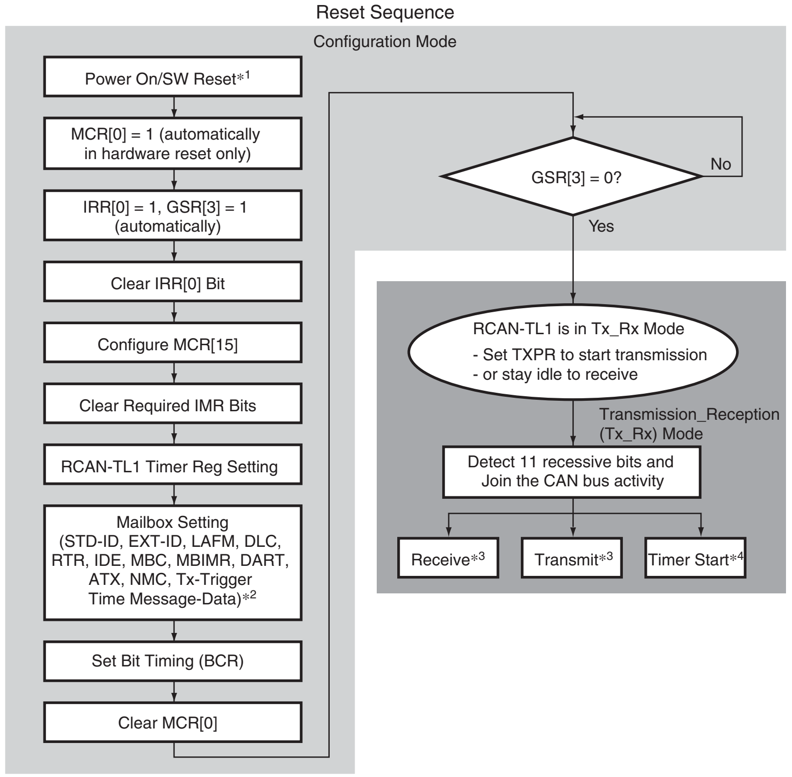
When message error occurs, IRR13 is set in all test modes.

18.4.2 Configuration of RCAN-TL1

RCAN-TL1 is considered in configuration mode or after a H/W (Power On Reset)/S/W (MCR[0]) reset or when in Halt mode. In both conditions RCAN-TL1 cannot join the CAN Bus activity and configuration changes have no impact on the traffic on the CAN Bus.

- After a Reset request

The following sequence must be implemented to configure the RCAN-TL1 after (S/W or H/W) reset. After reset, all the registers are initialized, therefore, RCAN-TL1 needs to be configured before joining the CAN bus activity. Please read the notes carefully.


Reset Sequence
Configuration Mode
Power On/SW Reset*1
MCR[0] = 1 (automatically in hardware reset only)
IRR[0] = 1, GSR[3] = 1 (automatically)
Clear IRR[0] Bit
Configure MCR[15]
Clear Required IMR Bits
RCAN-TL1 Timer Reg Setting
Mailbox Setting (STD-ID, EXT-ID, LAFM, DLC, RTR, IDE, MBC, MBIMR, DART, ATX, NMC, Tx-Trigger Time Message-Data)*2
Set Bit Timing (BCR)
Clear MCR[0]
GSR[3] = 0?
No
Yes
RCAN-TL1 is in Tx_Rx Mode
- Set TXPR to start transmission
- or stay idle to receive
Transmission_Reception (Tx_Rx) Mode
Detect 11 recessive bits and Join the CAN bus activity
Receive*3
Transmit*3
Timer Start*4

Notes: 1. SW reset could be performed at any time by setting MCR[0] = 1.
2. Mailboxes are comprised of RAMs, therefore, please initialize all mailboxes. If any data is read from a mailbox without initialization, a parity error may occur.
3. If there is no TXPR set, RCAN-TL1 will receive the next incoming message. If there is a TXPR(s) set, RCAN-TL1 will start transmission of the message and will be arbitrated by the CAN bus. If it loses the arbitration, it will become a receiver.
4. Timer can be started at any time after the Timer Control regs and Tx-Trigger Time are set.

Figure 18.14 Reset Sequence

- Halt mode

When RCAN-TL1 is in Halt mode, it cannot take part to the CAN bus activity. Consequently the user can modify all the requested registers without influencing existing traffic on the CAN Bus. It is important for this that the user waits for the RCAN-TL1 to be in halt mode before to modify the requested registers - note that the transition to Halt Mode is not always immediate (transition will occurs when the CAN Bus is idle or in intermission). After RCAN-TL1 transit to Halt Mode, GSR4 is set.

Once the configuration is completed the Halt request needs to be released. RCAN-TL1 will join CAN Bus activity after the detection of 11 recessive bits on the CAN Bus.

- Sleep mode

When RCAN-TL1 is in sleep mode the clock for the main blocks of the IP is stopped in order to reduce power consumption. Only the following user registers are clocked and can be accessed: MCR, GSR, IRR and IMR. Interrupt related to transmission (TXACK and ABACK) and reception (RXPR and RFPR) cannot be cleared when in sleep mode (as TXACK, ABACK, RXPR and RFPR are not accessible) and must to be cleared beforehand.

The following diagram shows the flow to follow to move RCAN-TL1 into sleep mode.

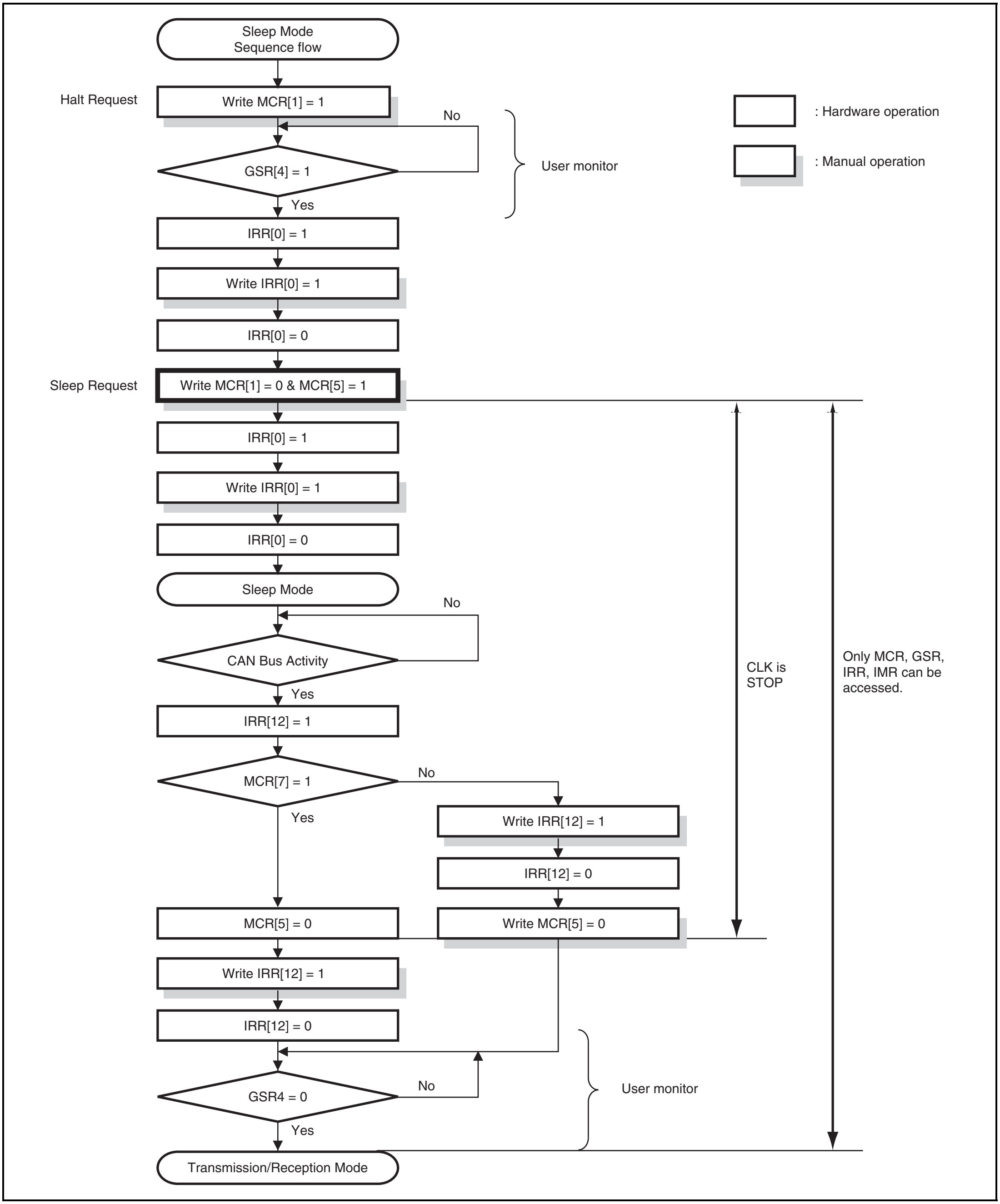


Figure 18.15 shows allowed state transitions.

- Please don't set MCR5 (Sleep Mode) without entering Halt Mode.
- After MCR1 is set, please don't clear it before GSR4 is set and RCAN-TL1 enters Halt Mode.

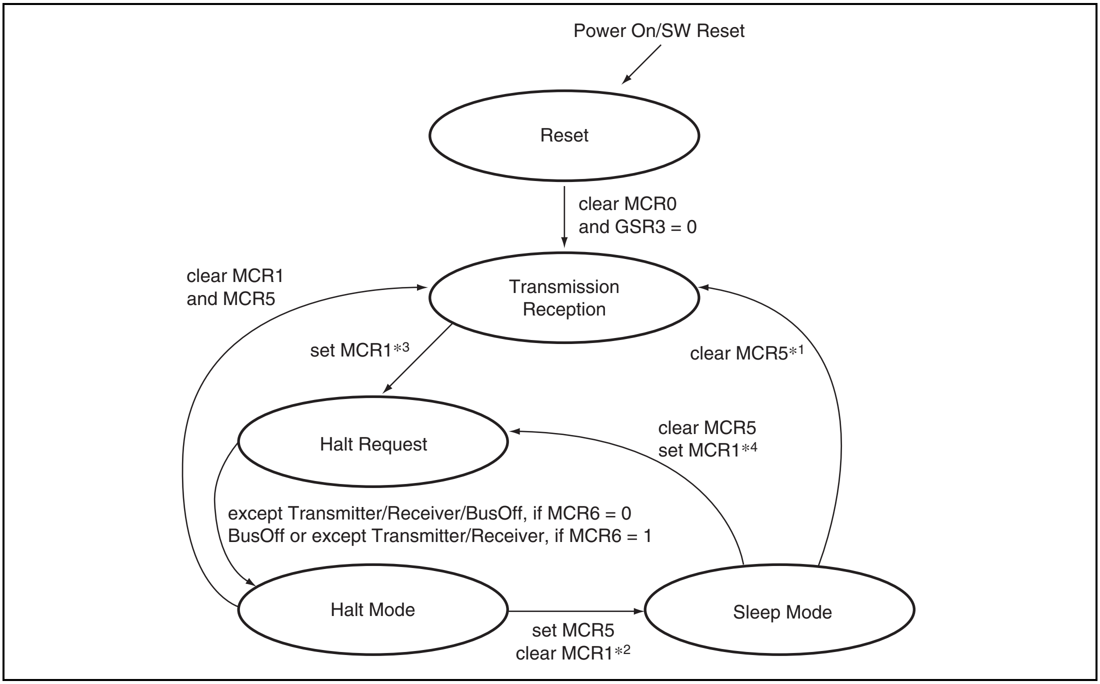


Figure 18.15 Halt Mode/Sleep Mode

Notes: 1. MCR5 can be cleared by automatically by detecting a dominant bit on the CAN Bus if MCR7 is set or by writing '0'.

2. MCR1 is cleared in SW. Clearing MCR1 and setting MCR5 have to be carried out by the same instruction.

3. MCR1 must not be cleared in SW, before GSR4 is set. MCR1 can be set automatically in HW when RCAN-TL1 moves to Bus Off and MCR14 and MCR6 are both set.

4. When MCR5 is cleared and MCR1 is set at the same time, RCAN-TL1 moves to Halt Request. Right after that, it moves to Halt Mode with no reception/transmission.

The following table shows conditions to access registers.

	RCAN-TL1 Registers								
Status Mode	**MCR GSR**	**IRR IMR**	**BCR**	**MBIMR timer TT_register**	**Flag_ register**	**Mailbox (ctrl0, LAFM)**	**Mailbox (data)**	**Mailbox (ctrl1)**	**Mailbox Trigger Time TT control**
Reset	yes	yes	yes	yes	yes	yes	yes	yes	yes
Transmission Reception Halt Request	yes	yes	no*1	yes	yes	no*1 yes*2	yes*2	no*1 yes*2	yes*2
Halt	yes	yes	no*1	yes	yes	yes	yes	yes	yes
Sleep	yes	yes	no	no	no	no	no	no	no

Notes: 1. No hardware protection.
2. When TXPR is not set.

18.4.3 Message Transmission Sequence

- Message Transmission Request

The following sequence is an example to transmit a CAN frame onto the bus. As described in the previous register section, please note that IRR8 is set when one of the TXACK or ABACK bits is set, meaning one of the Mailboxes has completed its transmission or transmission abortion and is now ready to be updated for the next transmission, whereas, the GSR2 means that there is currently no transmission request made (No TXPR flags set).

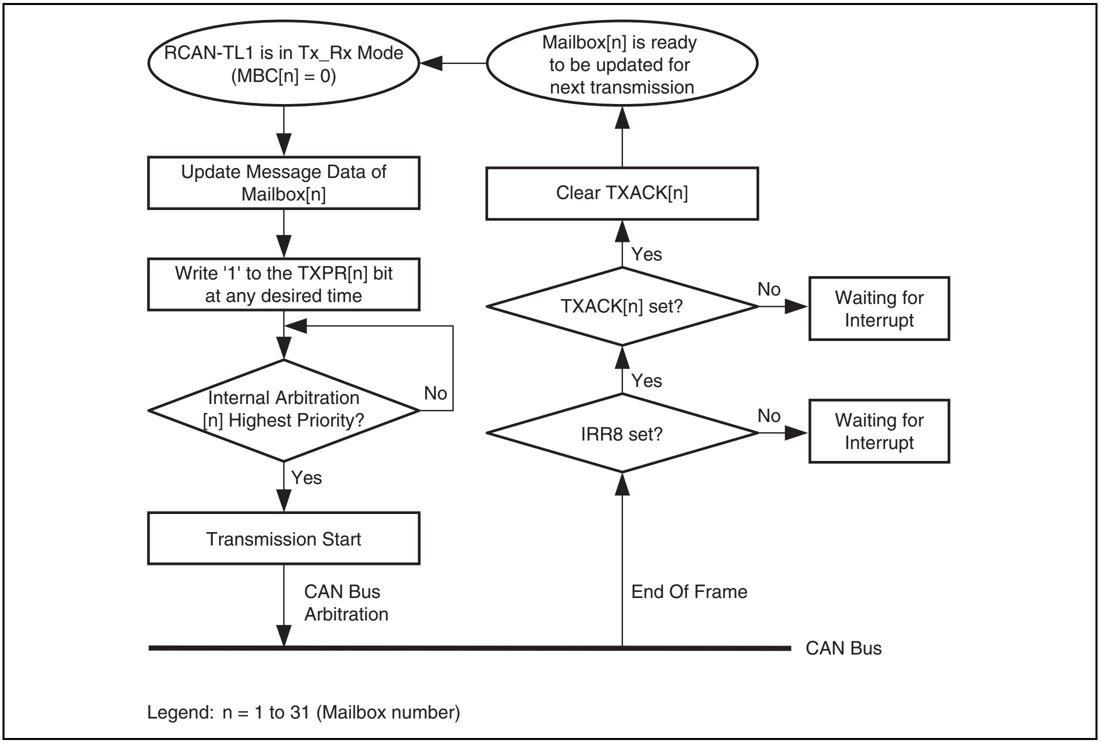

RCAN-TL1 is in Tx_Rx Mode (MBC[n] = 0)
Mailbox[n] is ready to be updated for next transmission
Update Message Data of Mailbox[n]
Clear TXACK[n]
Yes
Write '1' to the TXPR[n] bit at any desired time
TXACK[n] set?
No
Waiting for Interrupt
Yes
Internal Arbitration [n] Highest Priority?
No
IRR8 set?
No
Waiting for Interrupt
Yes
Transmission Start
CAN Bus Arbitration
End Of Frame
CAN Bus
Legend: n = 1 to 31 (Mailbox number)

Figure 18.16 Transmission request

- Internal Arbitration for transmission

The following diagram explains how RCAN-TL1 manages to schedule transmission-requested messages in the correct order based on the CAN identifier. 'Internal arbitration' picks up the highest priority message amongst transmit-requested messages.

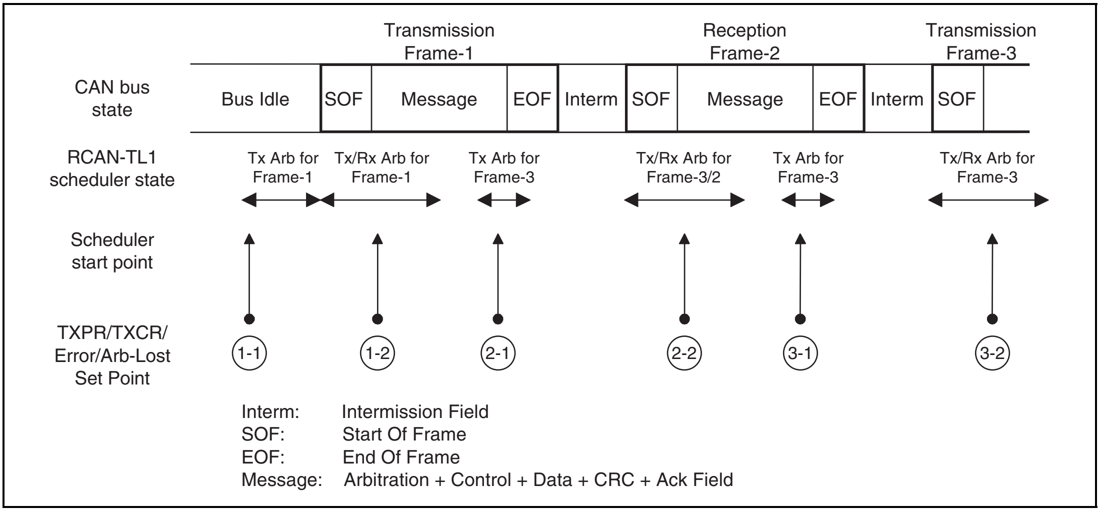


Figure 18.17 Internal Arbitration for transmission

The RCAN-TL1 has two state machines. One is for transmission, and the other is for reception.

1-1: When a TXPR bit(s) is set while the CAN bus is idle, the internal arbitration starts running immediately and the transmission is started.

1-2: Operations for both transmission and reception starts at SOF. Since there is no reception frame, RCAN-TL1 becomes transmitter.

2-1: At crc delimiter, internal arbitration to search next message transmitted starts.

2-2: Operations for both transmission and reception starts at SOF. Because of a reception frame with higher priority, RCAN-TL1 becomes receiver. Therefore, Reception is carried out instead of transmitting Frame-3.

3-1: At crc delimiter, internal arbitration to search next message transmitted starts.

3-2: Operations for both transmission and reception starts at SOF. Since a transmission frame has higher priority than reception one, RCAN-TL1 becomes transmitter.

Internal arbitration for the next transmission is also performed at the beginning of each error delimiter in case of an error is detected on the CAN Bus. It is also performed at the beginning of error delimiters following overload frame.

As the arbitration for transmission is performed at CRC delimiter, in case a remote frame request is received into a Mailbox with ATX = 1 the answer can join the arbitration for transmission only at the following Bus Idle, CRC delimiter or Error Delimiter.

Depending on the status of the CAN bus, following the assertion of the TXCR, the corresponding Message abortion can be handled with a delay of maximum 1 CAN Frame.

(1) Time Triggered Transmission

RCAN-TL1 offers a H/W support to perform communication in Time Trigger mode in line with the emerging ISO-11898-4 Level 1 Specification.

This section reports the basic procedures to use this mode.

- Setting Time Trigger Mode

In order to set up the time trigger mode the following settings need to be used.

- — CMAX in CMAX_TEW must be programmed to a value different from 3'b111.
- — Bit 15 in TTCR0 has to be set, to start TCNTR.
- — Bit 6 in TTCR0 has to be cleared to prevent TCNTR from being cleared after a match.
- — DART in Mailboxes used for time-triggered transmission cannot be used, since for Time Triggered Mailboxes, TXPR is not cleared to support periodic transmission.

- Roles of Registers

The user registers of RCAN-TL1 can be used to handle the main functions requested by the TTCAN standard.

TCNTR	Local Time
RFMK	Ref_Mark
CYCTR	Cycle Time = TCNTR - RFMK
RFTROFF	Ref_Trigger_Offset for Mailbox-30
Mailbox-31	Mailbox dedicated to the reception of time reference message
Mailbox-30	Mailbox dedicated to the transmission of time reference message when working as a potential time master
Mailbox-29 to 24	Mailboxes supporting time-triggered transmission
Mailbox-23 to 16	Mailboxes supporting reception without timestamp (may also be implemented as Mailboxes supporting Event Triggered transmission)
Mailbox-15 to 0	Mailboxes supporting reception with timestamp timestamp (may also be implemented as Mailboxes supporting Event Triggered transmission)
Tx-Trigger Time	Time_Mark to specify when a message should be transmitted

CMAX Specifies the maximum number of basic cycles when working as potential time master

TEW Specify the width of Tx_Enable

TCMR0 Init_Watch_Trigger (compare match with Local Time)

TCMR1 Compare match with Cycle Time to monitor users-specified events

TCMR2 Watch_Trigger (compare match with Cycle Time). This can be programmed to abort all pending transmissions

TTW Specifies the attribute of a time window used for transmission

TTTSEL Specifies the next Mailbox waiting for transmission

- Time Master/Time Slave

RCAN-TL1 can be programmed to work as a potential time master of the network or as a time slave. The following table shows the settings and the operation automatically performed by RCAN-TL1 in each mode.

mode	requested setting	function
Time Slave	TXPR[30] = 0 & MBC[30]!= 3'b000 & CMAX!= 3'b111 & MBC[31] = 3'b011	TCNTR is sampled at each SOF detected on the CAN Bus and stored into an internal register. When a valid Time Reference Message is received into Mailbox-31 the value of TCNTR (stored at the SOF) is copied into Ref_Mark. CCR embedded in the received Reference Message is copied to CCR. If Next_is_Gap = 1, IRR13 is set.
(Potential) Time Master	TXPR[30] = 1 & MBC[30] = 3'b000 & DLC[30] > 0 & CMAX!= 3'b111 & MBC[31] = 3'b011	Two cases are covered: (1) When a valid Time Reference message is received into Mailbox-31 the value of TCNTR stored into an internal register at the SOF is copied into Ref_Mark. CCR embedded in the received Reference Message is copied to CCR. If Next_is_Gap = 1, IRR13 is set. (2) When a Time Reference message is transmitted from Mailbox-30 the value of TCNTR stored into an internal register at the SOF is copied into Ref_Mark. CCR is incremented when TTT of Mailbox-30 matches with CYCTR. CCR is embedded into the first data byte of the time reference message { Data0[7:6], CCR[5:0] }

- Setting Tx-Trigger Time

The Tx-Trigger Time(TTT) must be set in ascending order shown below, and the difference between them has to satisfy the following expressions. TEW in the following expressions is the register value.

TTT (Mailbox-24) < TTT (Mailbox-25) < TTT (Mailbox-26) < TTT (Mailbox-27) < TTT (Mailbox-28) < TTT (Mailbox-29) < TTT (Mailbox-30)

and

TTT (Mailbox-n) – TTT (Mailbox- n-1) > TEW + the maximum frame length + 9
n = 25 to 30

TTT (Mailbox-24) to TTT (Mailbox-29) correspond to Time_Marks, and TTT (Mailbox-30) corresponds to Time_Ref showing the length of a basic cycle, respectively when working as potential time master.

The above limitation is not applied to mailboxes which are not set as time-triggered transmission.

Important: Because of limitation on setting Tx-Trigger Time, only one Mailbox can be assigned to one time window.

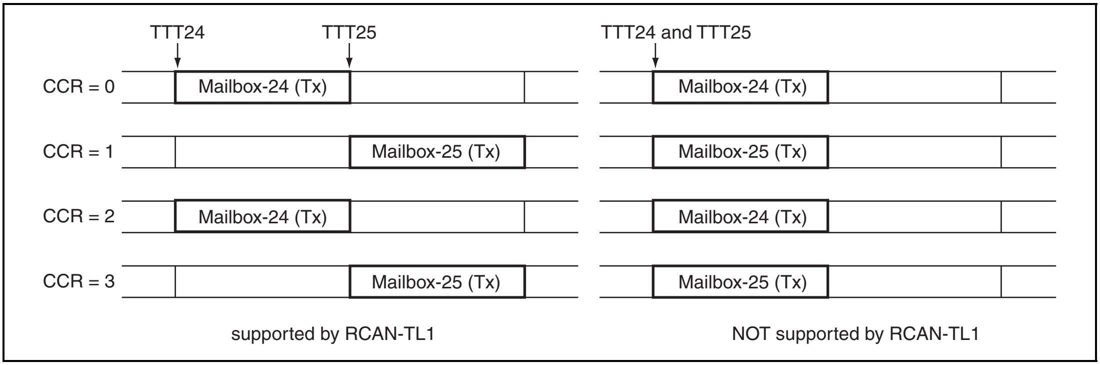


Figure 18.18 Limitation on Tx-Trigger Time

The value of TCMR2 as Watch_Trigger has to be larger than TTT(Mailbox-30), which shows the length of a basic cycle.

Figure 18.19 and Figure 18.20 show examples of configurations for (Potential) Time Master and Time Slave. "L" in diagrams shows the length in time of the time reference messages.

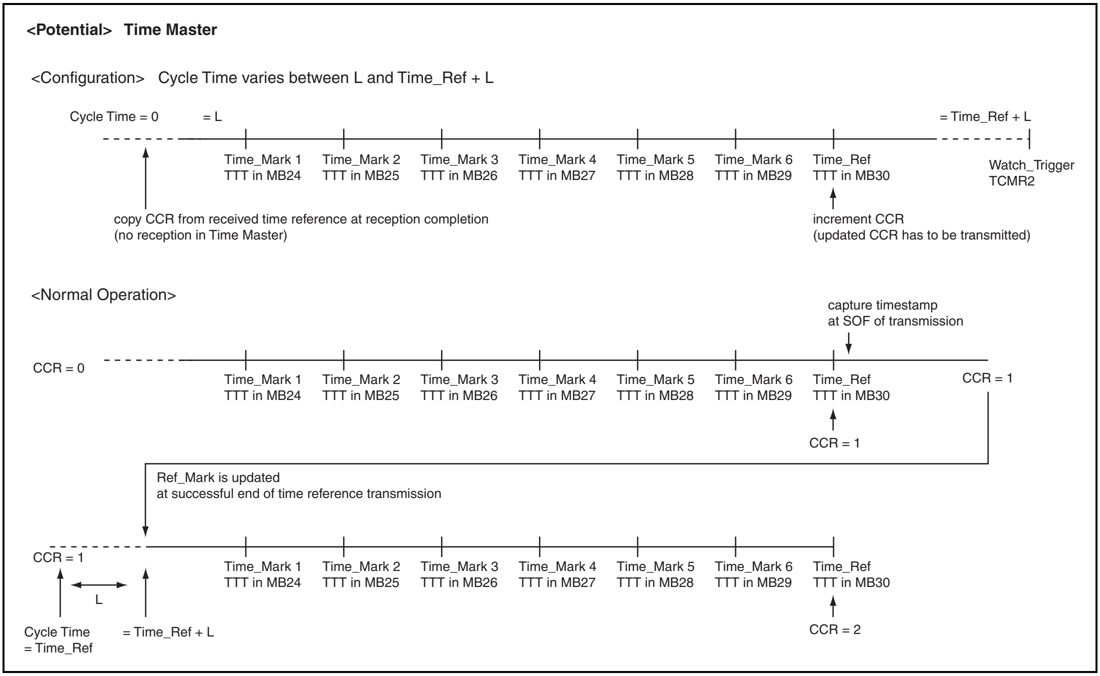


Figure 18.19 (Potential) Time Master

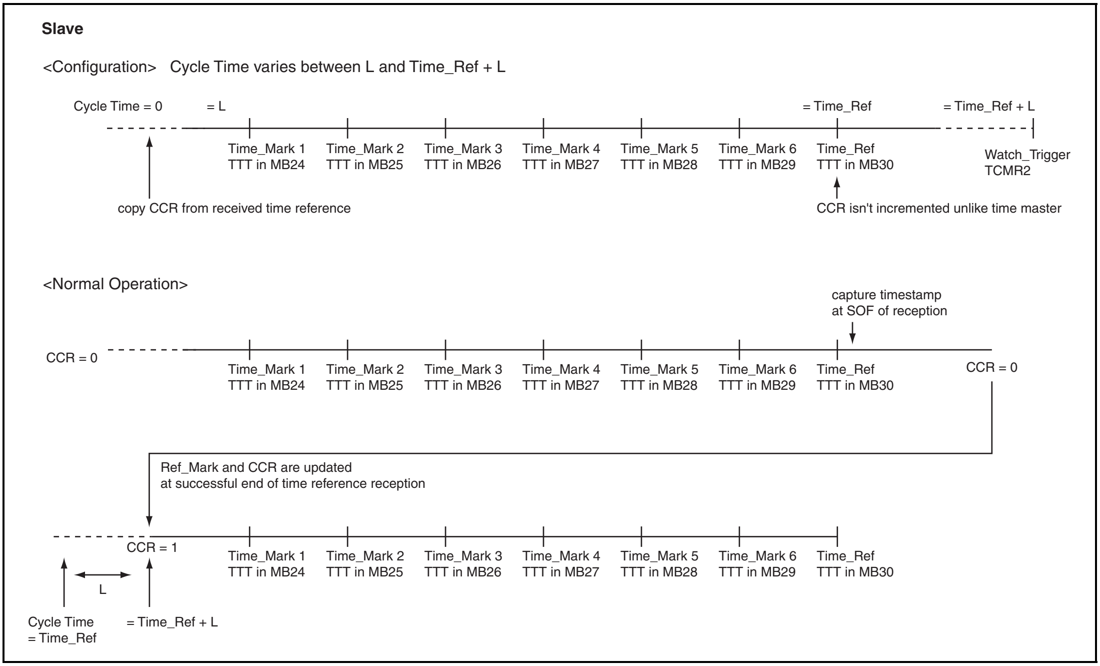


Figure 18.20 Time Slave

- Function to be implemented by software

Some of the TTCAN functions need to be implemented in software. The main details are reported hereafter. Please refer to ISO-11898-4 for more details.

— Change from Init_Watch_Trigger to Watch_Trigger

RCAN-TL1 offers the two registers TCMR0 and TCMR2 as H/W support for Init_Watch_Trigger and Watch_Trigger respectively. The SW is requested to enable TCMR0 and disable TCMR2 up to the first reference message is detected on the CAN Bus and then disable TCMR0 and enable TCMR2.- Schedule Synchronization state machine.

Only reception of Next_is_Gap interrupt is supported. The application needs to take care of stopping all transmission at the end of the current basic cycle by setting the related TXCR flags.Master-Slave Mode control.

Only automatic cycle time synchronization and CCR increment is supported.

— Message status count

Software has to count scheduling errors for periodic messages in exclusive windows.

- Message Transmission Request for Time Triggered communication

When the Time Triggered mode is used communications must fulfils the ISO11898-4 requirements.

The following procedure should be used.

- Send RCAN-TL1 to reset or halt mode
- Set TCMR0 to the Init_Watch_Trigger (0xFFFF)
- Enable TCMR0 compare match setting bit 10 of TTCR0
- Set TCMR2 to the specified Watch_Trigger value
- Keep TCMR2 compare match disabled by keeping cleared the bit 12 of TTCR0
- Set CMAX to the requested value (different from 111 bin)
- Set TEW to the requested value
- Configure the necessary Mailboxes for Time Trigger transmission and reception
- Set LAFM for the 3 LSBs of Mailbox 31
- Configure MCR, BCR1 and BCR0 to the requested values
- If working as a potential time master:
  - Set RFTROFF to the requested Init_Ref_Offset value
  - Set TXPR for Mailbox 30
  - Write H'4000 into TTTSEL
- Enable the TCNTR timer through the bit 15 of TTCR0
- Move to Transmission_Reception mode
- Wait for the reception or transmission of a valid reference message or for TCMR0 match
- If the local time reaches the value of TCMR0 the Init_Watch_Trigger is reached and the application needs to set TXCR for Mailbox 30 and start again
- If the reference message is transmitted (TXACK[30] is set) set RFTROFF to zero
- If a valid reference message is received (RXPR[31] is set) then:
  - If 3 LSBs of ID of Mailbox 31 have high priority than the 3 LSBs of Mailbox 30 (if working as potential time master) keep RFTROFF to Init_Ref_Offset
  - If 3 LSBs of ID of Mailbox 31 have lower priority than the 3 LSBs of Mailbox 30 (if working as potential time master) decrement by 1 the value in RFTROFF
- Disable TCMR0 compare match by clearing bit 10 of TTCR0
- Enable TCMR2 compare match by setting bit 12 of TTCR0
- Only after two reference messages have been detected on the CAN Bus (transmitted or received) can the application set TXPR for the other Time Triggered Mailboxes.

If, at any time, a reference message cannot be detected on the CAN Bus, and the cycle time CYCTR reaches TCMR2, RCAN-TL1 automatically aborts all pending transmissions (including the Reference Message).

The following is the sequence to request further transmission in Time Triggered mode.

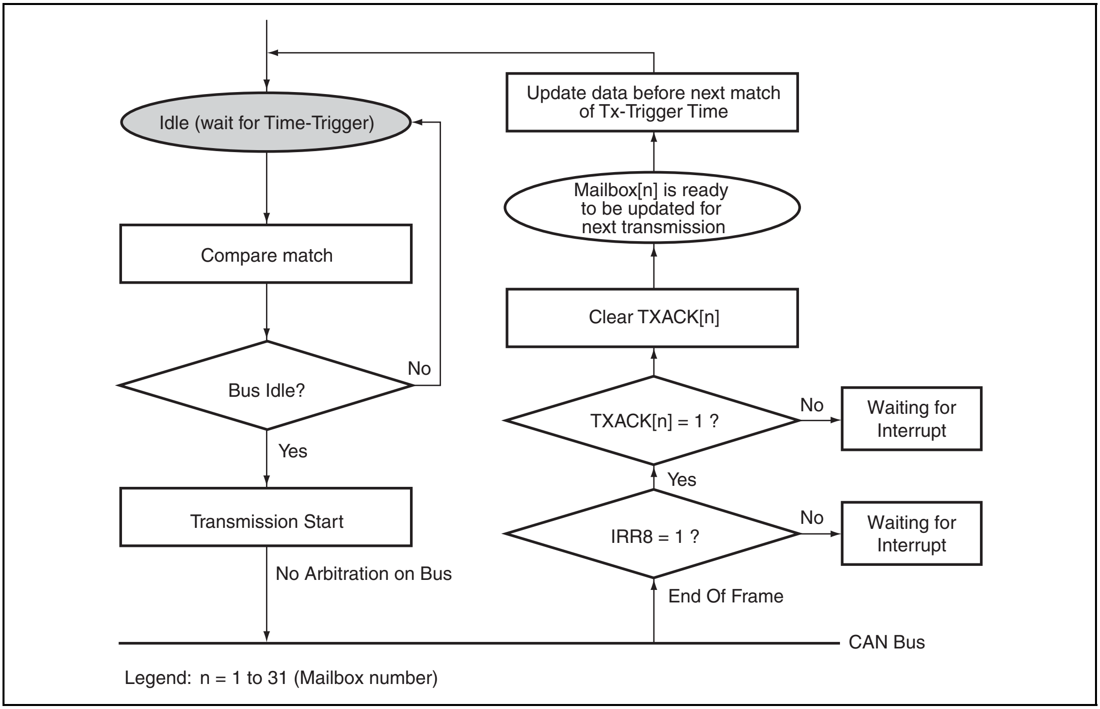


Figure 18.21 Message transmission request

S/W has to ensure that a message is updated before a Tx trigger for transmission occurs.

When the CYCTR reaches to TTT (Tx-Trigger Time) of a Mailbox and CCR matches with the programmed cycle for transmission, RCAN-TL1 immediately transfers the message into the Tx buffer. At this point, RCAN-TL1 will attempt a transmission within the specified Time Enable Window. If RCAN-TL1 misses this time slot, it will suspend the transmission request up to the next Tx Trigger, keeping the corresponding TXPR bit set to '1' if the transmission is periodic (Mailbox-24 to 30). There are three factors that may cause RCAN-TL1 to miss the time slot –

1. The CAN bus currently used
2. An error on the CAN bus during the time triggered message transmission
3. Arbitration loss during the time triggered message transmission

In case of Merged Arbitrating Window the slot for transmission goes from the Tx_Trig of the Mailbox opening the Window (TTW = 10 bin) to the end to the TEW of the Mailbox closing the Window (TTW = 11 bin).The TXPR can be modified at any time. RCAN-TL1 ensures the transmission of Time Triggered messages is always scheduled correctly. However, in order to guarantee the correct schedule, there are some important rules that are :

— TTT (Tx Trigger Time) can be modified during configuration mode.

— TTT cannot be set outside the range of Time_Ref, which specifies the length of basic cycle. This could cause a scheduling problem.

— TXPR is not automatically cleared for periodic transmission. If a periodic transmission needs to be cancelled, the corresponding TXCR bit needs to be set by the application.

- Example of Time Triggered System

The following diagram shows a simple example of how time trigger system works using RCAN-TL1 in time slave mode.

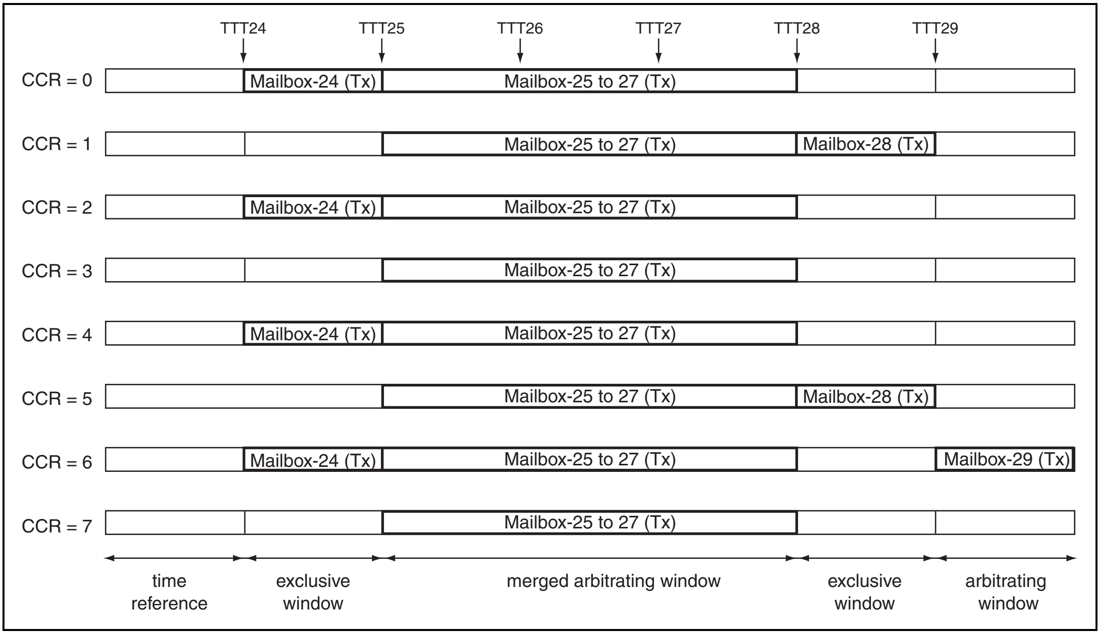


Figure 18.22 Example of Time trigger system as Time Slave

The following settings were used in the above example:

	rep_factor (register)	Offset	TTW[1:0]	MBC[2:0]
Mailbox-24	3'b001	6'b000000	2'b00	3'b000
Mailbox-25	3'b000	6'b000000	2'b10	3'b000
Mailbox-26	3'b000	6'b000000	2'b10	3'b000
Mailbox-27	3'b000	6'b000000	2'b11	3'b000
Mailbox-28	3'b010	6'b000001	2'b00	3'b000
Mailbox-29	3'b011	6'b000110	2'b01	3'b000
Mailbox-30	—	—	—	3'b111
Mailbox-31	—	—	—	3'b011

CMAX = 3'b011, TXPR[30] = 0

During merged arbitrating window, request by time-triggered transmission is served in the way of FCFS (First Come First Served). For example, if Mailbox-25 cannot be transmitted between Tx-Trigger Time 25 (TTT25) and TTT26, Mailbox-25 has higher priority than Mailbox-26 between TTT26 and 28.

MBC needs to be set into 3'b111, in order to disable time-triggered transmission. If RCAN-TL1 is Time Master, MBC[30] has to be 3'b000 and time reference window is automatically recognized as arbitrating window.

- Timer Operation

Figure 18.23 shows the timing diagram of the timer. By setting Tx-Trigger Time = n, time trigger transmission starts between CYCTR = n + 2 and
CYCTR = n + 3.

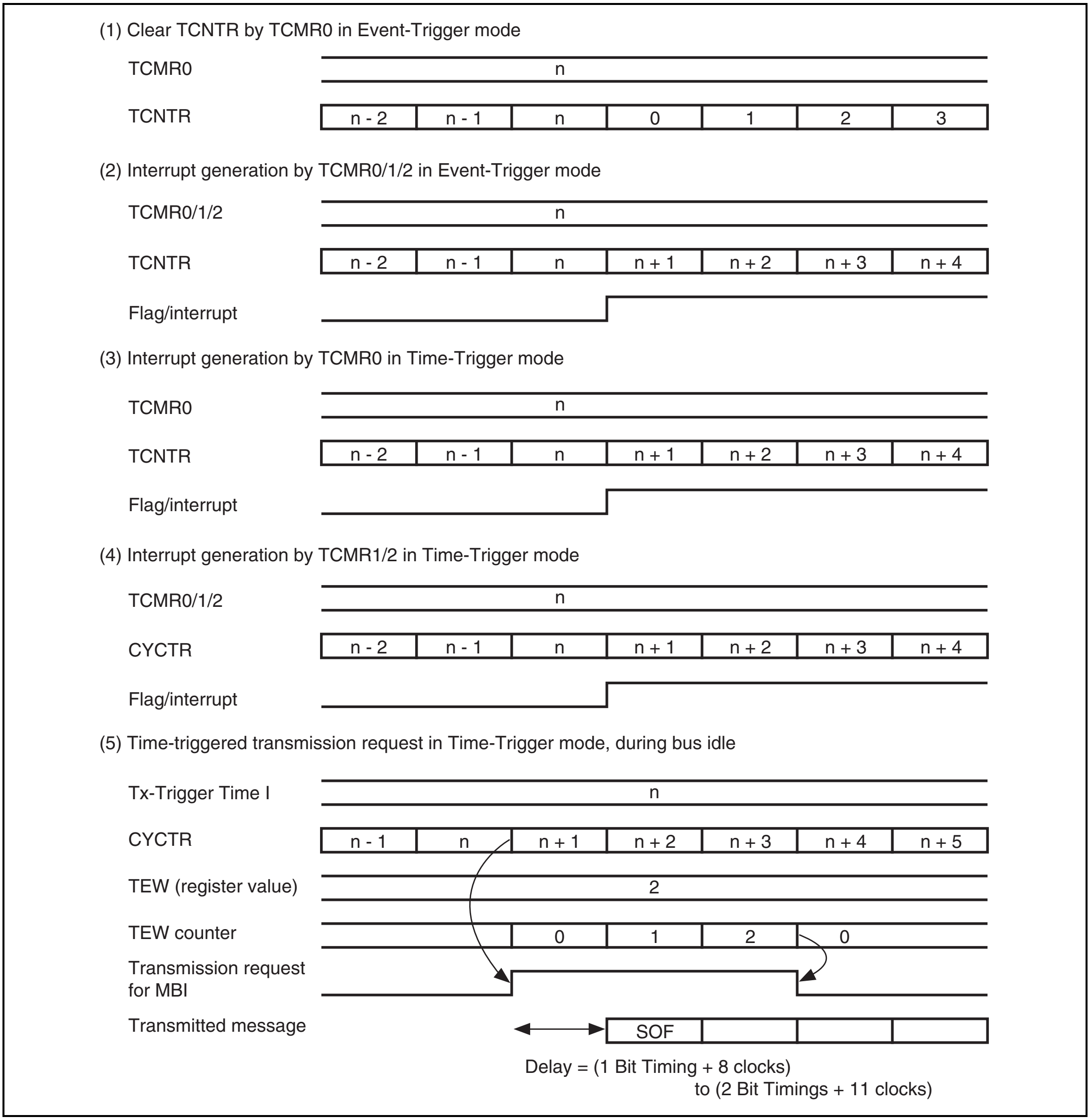

(1) Clear TCNTR by TCMR0 in Event-Trigger mode
TCMR0
n
TCNTR
n - 2
n - 1
n
0
1
2
3
(2) Interrupt generation by TCMR0/1/2 in Event-Trigger mode
TCMR0/1/2
n
TCNTR
n - 2
n - 1
n
n + 1
n + 2
n + 3
n + 4
Flag/interrupt
(3) Interrupt generation by TCMR0 in Time-Trigger mode
TCMR0
n
TCNTR
n - 2
n - 1
n
n + 1
n + 2
n + 3
n + 4
Flag/interrupt
(4) Interrupt generation by TCMR1/2 in Time-Trigger mode
TCMR0/1/2
n
CYCTR
n - 2
n - 1
n
n + 1
n + 2
n + 3
n + 4
Flag/interrupt
(5) Time-triggered transmission request in Time-Trigger mode, during bus idle
Tx-Trigger Time I
n
CYCTR
n - 1
n
n + 1
n + 2
n + 3
n + 4
n + 5
TEW (register value)
2
TEW counter
0
1
2
0
Transmission request for MBI
Transmitted message
SOF
Delay = (1 Bit Timing + 8 clocks)
to (2 Bit Timings + 11 clocks)

Figure 18.23 Timing Diagram of Timer

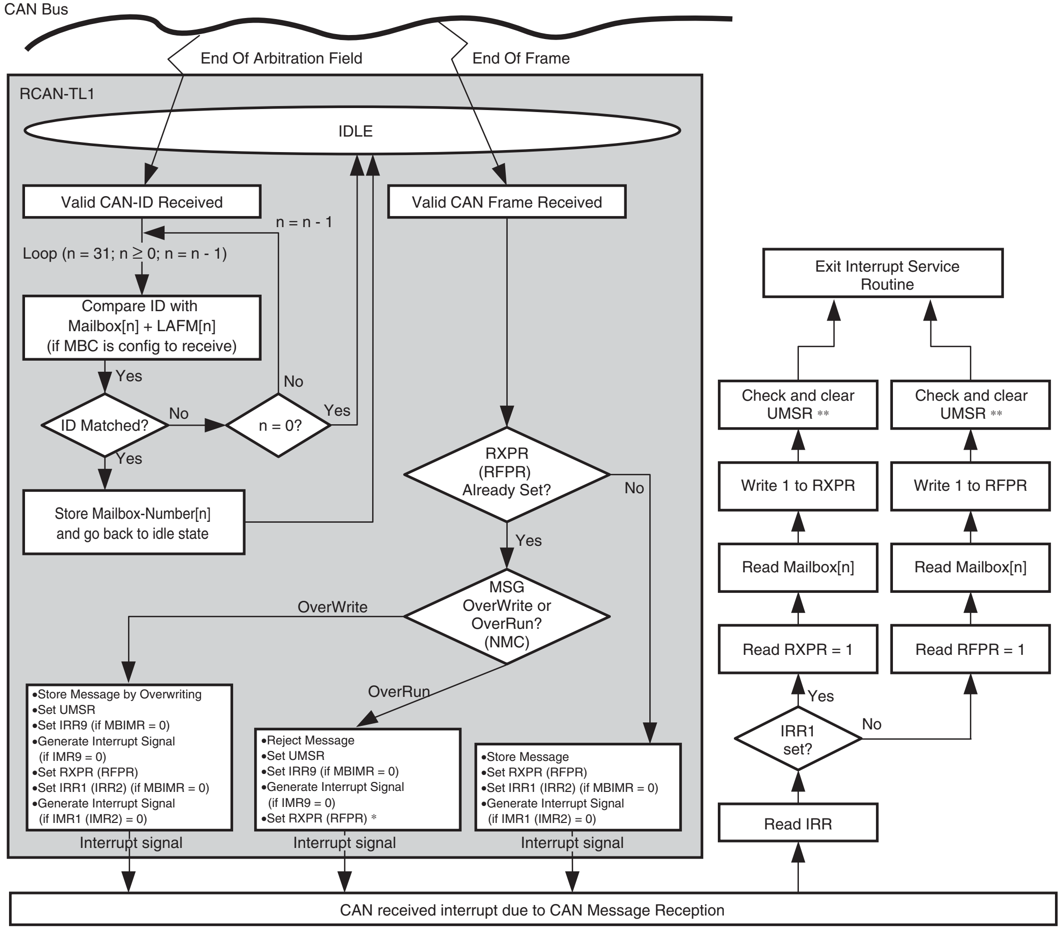
During merged arbitrating window, event-trigger transmission is served after completion of time-triggered transmission. For example, If transmission of Mailbox-25 is completed and CYCTR doesn't reach TTT26, event-trigger transmission starts based on message transmission priority specified by MCR2. TXPR of time-triggered transmission is not cleared after transmission completion, however, that of event-triggered transmission is cleared.

Note: that in the case that the TXPR is not set for the Mailbox which is assigned to close the Merged Arbitrating Window (MAW), then the MAW will still be closed (at the end of the TEW following the TTT of the assigned Mailbox.

Please refer to section 18.3.2, Mailbox Structure.

18.4.4 Message Receive Sequence

The diagram below shows the message receive sequence.



Legend: n = 0 to 31 (Mailbox number)

Notes: 1. Only if CPU clears RXPR/RFPR at the same time that UMSR is set in overrun, RXPR/RFPR may be set again even though the message has not been updated. TimeStamp may also be updated, however it can be read properly before clearing RXPR/RFPR.

2. In case overwrite configuration (NMC = 1) is used for the Mailbox n the message must be discarded when UMSR = 1, UMSR cleared and the full Interrupt Service Routine started again. In case of overrun configuration (NMC = 0) is used clear again RXPR/RFPR/UMSR when UMSR = 1 and consider the message obsolate.

Figure 18.24 Message receive sequence

When RCAN-TL1 recognises the end of the Arbitration field while receiving a message, it starts comparing the received identifier to the identifiers set in the Mailboxes, starting from Mailbox-31 down to Mailbox-0. It first checks the MBC if it is configured as a receive box, and reads LAFM, and reads the CAN-ID of Mailbox-31 (if configured as receive) to finally compare them to the received ID. If it does not match, the same check takes place at Mailbox-30 (if configured as receive). Once RCAN-TL1 finds a matching identifier, it stores the number of Mailbox-[n] into an internal buffer, stops the search, and goes back to idle state, waiting for the EndOfFrame (EOF) to come. When the 6th bit of EOF is notified by the CAN Interface logic, the received message is written or abandoned, depending on the NMC bit. No modification of configuration during communication is allowed. Entering Halt Mode is one of ways to modify configuration. If it is written into the corresponding Mailbox, including the CAN-ID, i.e., there is a possibility that the CAN-ID is overwritten by a different CAN-ID of the received message due to the LAFM used. This also implies that, if the identifier of a received message matches to ID + LAFM of 2 or more Mailboxes, the higher numbered Mailbox will always store the relevant messages and the lower numbered Mailbox will never receive messages. Therefore, the settings of the identifiers and LAFMs need to be carefully selected.

With regards to the reception of data and remote frames described in the above flow diagram the clearing of the UMSR flag after the reading of IRR is to detect situations where a message is overwritten by a new incoming message stored in the same mailbox (if its NMC = 1) while the interrupt service routine is running. If during the final check of UMSR a overwrite condition is detected the message needs to be discarded and read again.

In case UMSR is set and the Mailbox is configured for overrun (NMC = 0) the message is still valid, however it is obsolete as it is not reflecting the latest message monitored on the CAN Bus.

Please access the full Mailbox content before clearing the related RXPR/RFPR flag.

Please note that in the case a received remote frame is overwritten by a data frame, both the remote frame receive interrupt (IRR2) and data frame received interrupt (IRR1) and also the Receive Flags (RXPR and RFPR) are set. In an analogous way, the overwriting of a data frame by a remote frame, leads to setting both IRR2 and IRR1.

When a message is received and stored into a Mailbox all the fields of the data not received are stored as zero. The same applies when a standard frame is received. The extended identifier part (EXTID[17:0]) is written as zero.

18.4.5 Reconfiguration of Mailbox

When re-configuration of Mailboxes is required, the following procedures should be taken.

- Change configuration of transmit box

  Two cases are possible.

  - Change of ID, RTR, IDE, LAFM, Data, DLC, NMC, ATX, DART

    This change is possible only when MBC = 3'b000. Confirm that the corresponding TXPR is not set. The configuration (except MBC bit) can be changed at any time.

  - Change from transmit to receive configuration (MBC)

    Confirm that the corresponding TXPR is not set. The configuration can be changed only in Halt or reset state. Please note that it might take longer for RCAN-TL1 to transit to halt state if it is receiving or transmitting a message (as the transition to the halt state is delayed until the end of the reception/transmission), and also RCAN-TL1 will not be able to receive/transmit messages during the Halt state.

    In case RCAN-TL1 is in the Bus Off state the transition to halt state depends on the configuration of the bit 6 of MCR and also bit and 14 of MCR.

- Change configuration (ID, RTR, IDE, LAFM, Data, DLC, NMC, ATX, DART, MBC) of receiver box or Change receiver box to transmitter box

  The configuration can be changed only in Halt Mode.

  RCAN-TL1 will not lose a message if the message is currently on the CAN bus and RCAN-TL1 is a receiver. RCAN-TL1 will be moving into Halt Mode after completing the current reception. Please note that it might take longer if RCAN-TL1 is receiving or transmitting a message (as the transition to the halt state is delayed until the end of the reception/transmission), and also RCAN-TL1 will not be able to receive/transmit messages during the Halt Mode.

  In case RCAN-TL1 is in the Bus Off state the transition to halt mode depends on the configuration of the bit 6 and 14 of MCR.

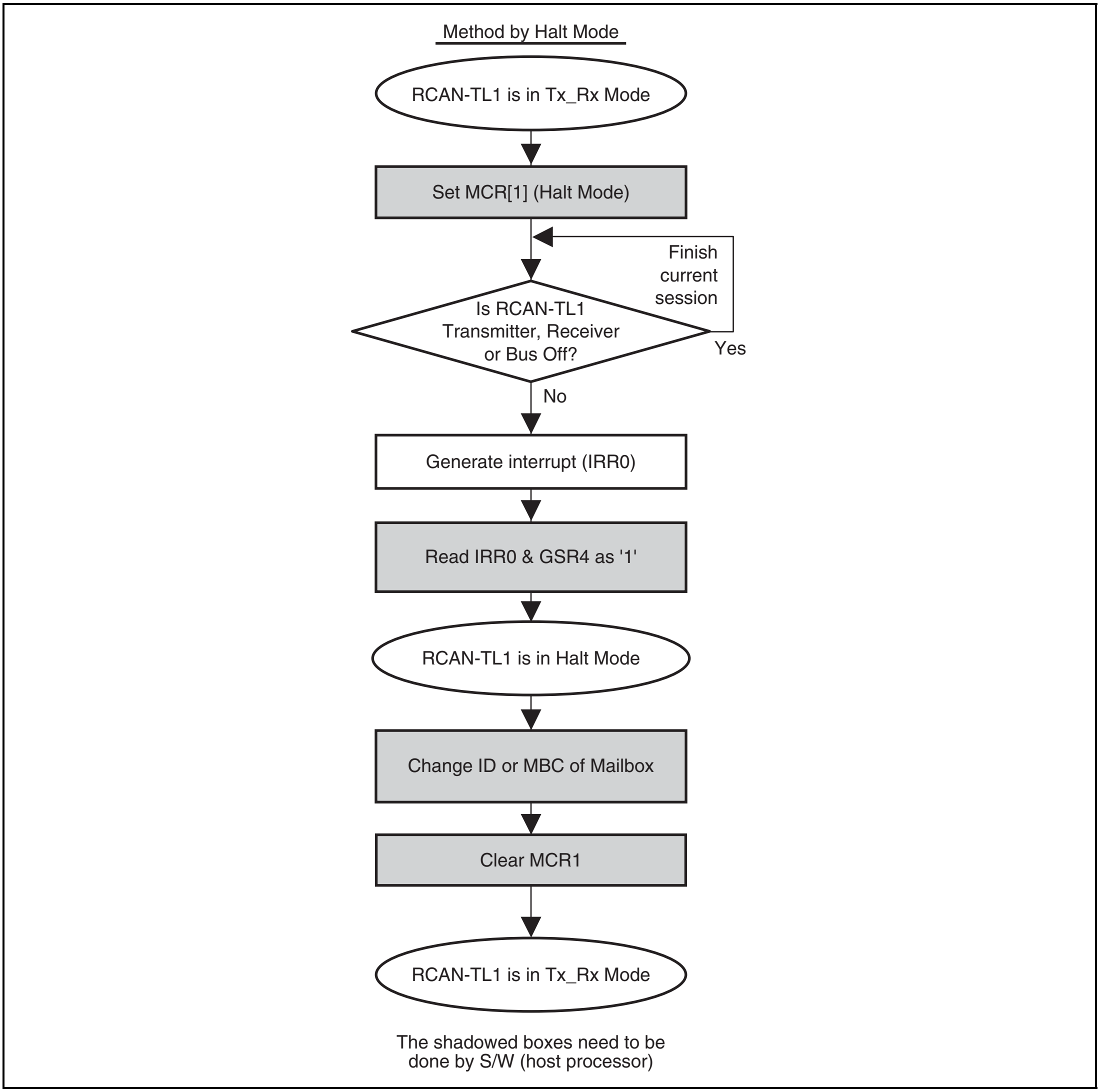

Method by Halt Mode
RCAN-TL1 is in Tx_Rx Mode
Set MCR[1] (Halt Mode)
Finish current session
Is RCAN-TL1 Transmitter, Receiver or Bus Off?
Yes
No
Generate interrupt (IRR0)
Read IRR0 & GSR4 as '1'
RCAN-TL1 is in Halt Mode
Change ID or MBC of Mailbox
Clear MCR1
RCAN-TL1 is in Tx_Rx Mode
The shadowed boxes need to be done by S/W (host processor)

Figure 18.25 Change ID of receive box or Change receive box to transmit box

18.5 Parity Detection

The RCAN-TL1 adds parity bits when data are written to a mailbox (RAM) and performs parity error detection when data are read from a mailbox. Interrupt generation upon parity error detection can be selected by setting the MBECR register.

- Parity addition: One parity bit is added to every 8 bits of data.
- Parity detection: Parity detection is performed in 32-bit units.

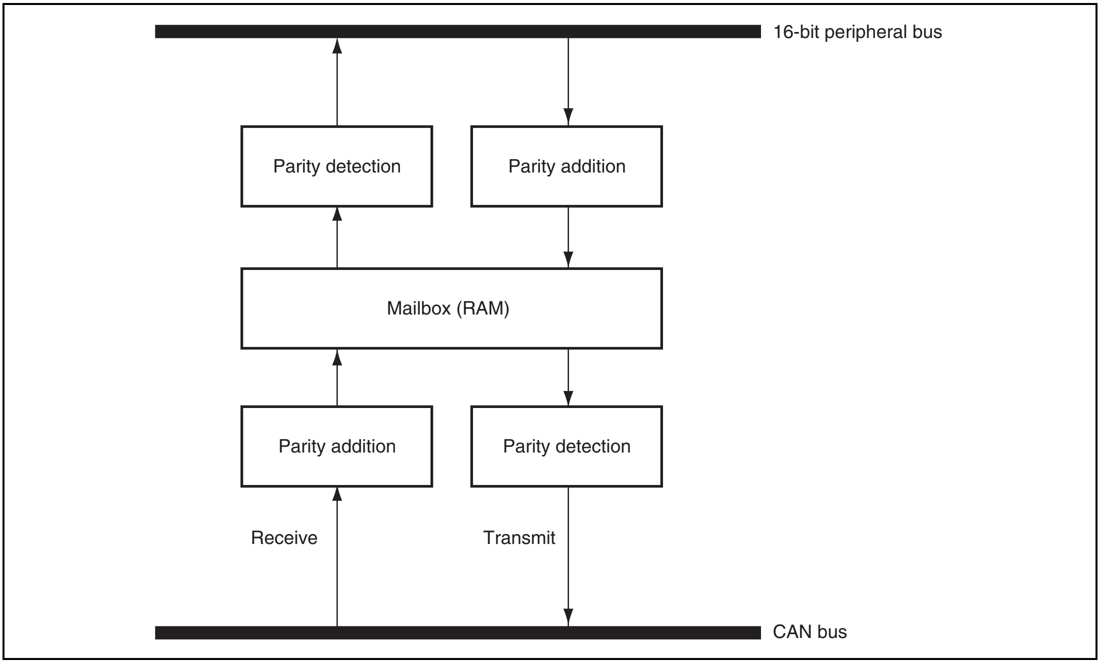


Figure 18.26 Parity Addition and Detection

18.6 Interrupt Sources

Table 18.2 lists the RCAN-TL1 interrupt sources. These sources can be masked. Masking is implemented using the mailbox interrupt mask registers (MBIMR) and interrupt mask register (IMR). For details on the interrupt vector of each interrupt source, see section 8, Interrupt Controller (INTC).

Table 18.2 RCAN-TL1-n*[1] Interrupt Sources

Interrupt	Description	Interrupt Flag	DMAC Activation	A-DMAC Activation
ERSn*[1]	Error Passive Mode (TEC ≥ 128 or REC ≥ 128)	IRR5	Not possible	Not possible
	Bus Off (TEC ≥ 256)/Bus Off recovery	IRR6		
	Error warning (TEC ≥ 96)	IRR3		
	Error warning (REC ≥ 96)	IRR4		
OVRn*[1]	Reset/halt/CAN sleep transition	IRR0		
	Overload frame transmission	IRR7		
	Unread message overwrite (overrun)	IRR9		
	Start of new system matrix	IRR10		
	TCMR2 compare match	IRR11		
	Bus activity while in sleep mode	IRR12		
	Timer overrun/Next_is_Gap reception/message error	IRR13		
	TCMR0 compare match	IRR14		
	TCMR1 compare match	IRR15		
RMn0*[1]*[2], RMn1*[1]*[2]	Data frame reception	IRR1*[3]	Possible*[4]	Possible*[5]
	Remote frame reception	IRR2*[3]		
SLEn*[1]	Message transmission/transmission disabled (slot empty)	IRR8	Not possible	Not possible
MBEn*[1]	Message buffer error	MBEF	Not possible	Not possible

Notes: 1. n = A, B, C, and D

2. RM0 is an interrupt generated by the remote request pending flag for mailbox 0 (RFPR0[0]) or the data frame receive flag for mailbox 0 (RXPR0[0]). RM1 is an interrupt generated by the remote request pending flag for mailboxes 1 to 31 (RFPR1/RFPR0) or the data frame receive flag for mailboxes 1 to 31 (RXPR1/RXPR0).

3. IRR1 is a data frame received interrupt flag for mailboxes 0 to 31, and IRR2 is a remote frame request interrupt flag for mailboxes 0 to 31.

4. The DMAC is activated only by an RMn0 interrupt.

5. The A-DMAC can be activated by an interrupt in any mailboxes.

18.7 DMAC Interface

The DMAC can be activated by the reception of a message in RCAN-TL1 mailbox 0. When DMAC transfer ends after DMAC activation has been set, flags of RXPR0 and RFPR0 are cleared automatically. An interrupt request due to a receive interrupt from the RCAN-TL1 cannot be sent to the CPU in this case. Figure 18.27 shows a DMAC transfer flowchart.

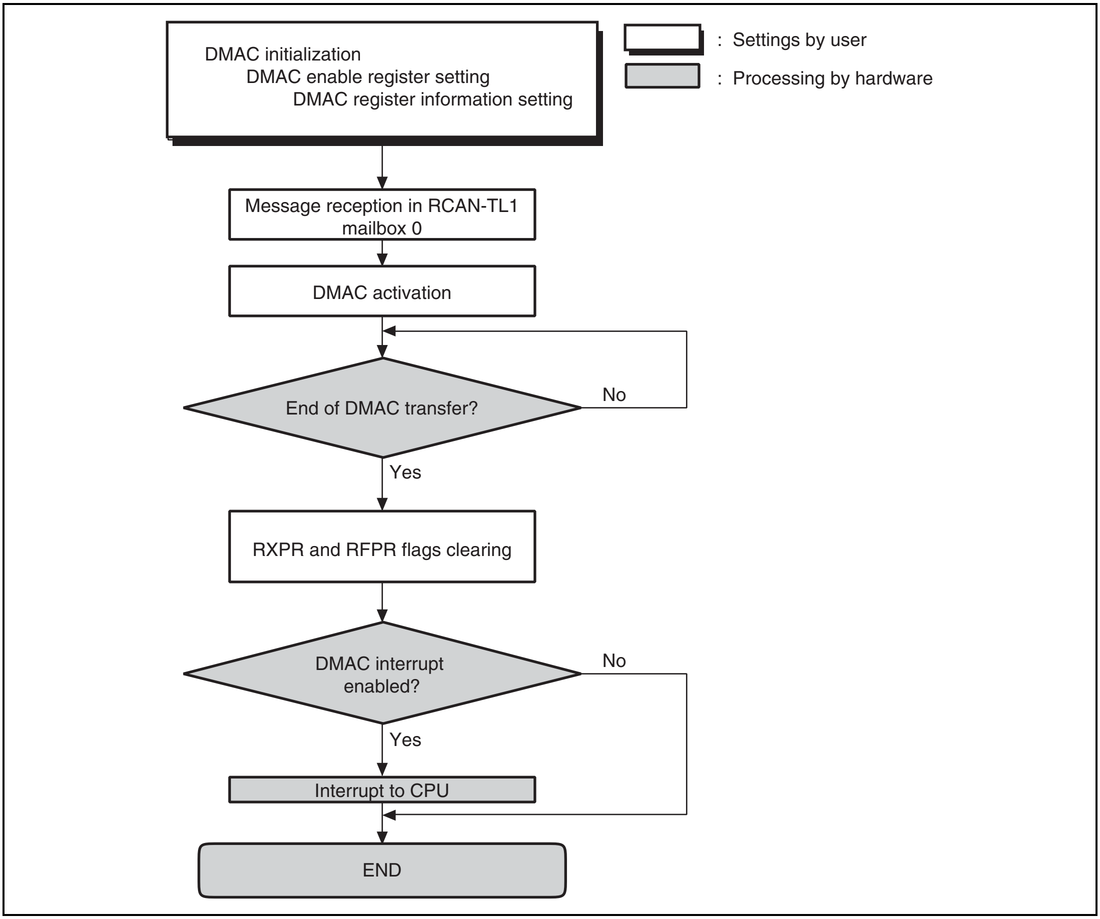


Figure 18.27 DMAC Transfer Flowchart

18.8 CAN Bus Interface

A bus transceiver IC is necessary to connect this LSI to a CAN bus. A Renesas HA13721 transceiver IC and its compatible products are recommended. Figure 18.28 shows a sample connection diagram.

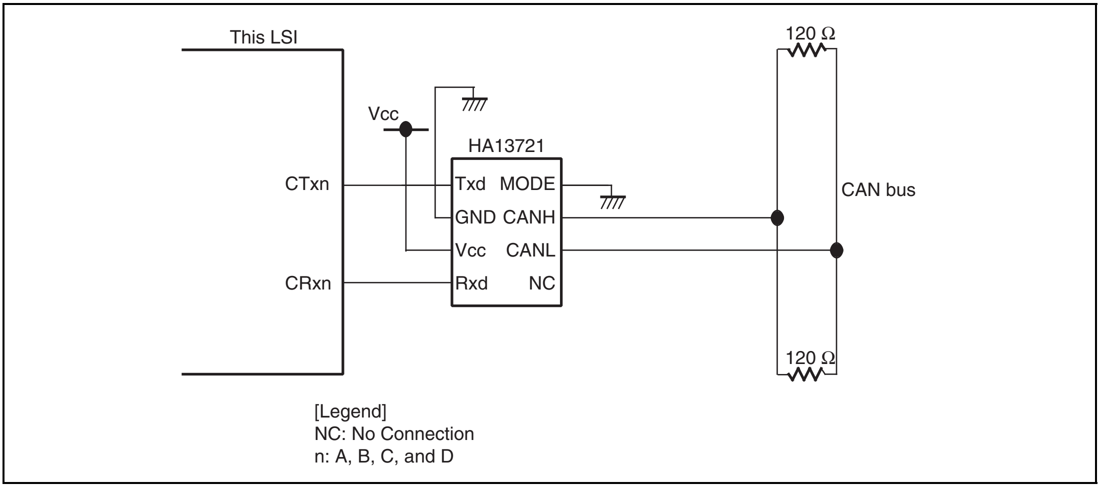


Figure 18.28 High-Speed CAN Interface Using HA13721

18.9 A-DMAC Interface

The A-DMAC can be activated by reception of a message in any mailboxes in any of four channels (RCAN-TL1-A, RCAN-TL1-B, RCAN-TL1-C, and RCAN-TL1-D).

When a message is received in a Mailbox after setting is completed, the A-DMAC is activated. The A-DMAC transfers the message data from the Mailbox to the RAM area and the flag in RXPR or RFPR is cleared to 0. At this time, an interrupt to the CPU is not generated by the RCAN-TL1 because it is masked by IMR.

For details on the setting of the A-DMAC, see section 12, Automotive Direct Memory Access Controller (A-DMAC).

18.10 Setting I/O Ports for RCAN-TL1

The I/O ports for the RCAN-TL1 must be specified before or during the configuration mode. For details on the settings of I/O ports, see section 23, Pin Function Controller (PFC). Three methods are available using four channels of the RCAN-TL1 in this LSI.

- Using RCAN-TL1 as a 4-channel module (channels A, B, C, and D). Each channel has 32 Mailboxes.
- Using RCAN-TL1 as a single channel module has 64 Mailboxes and as a 2-channels has 32 Mailboxes (channels A and B, channels C and D), or (channels A, B, channels C and D).
- Using RCAN-TL1 as a 2-channel module has 64 Mailboxes (channels A and B, channels C and D).
- Channels A, B, and C as a single channel has 96 Mailboxes and as a single channel module has 32 mailboxes (channel D).

When except for the first one mentioned above are used, see section 18.11.1, Notes on Port Setting for Multiple Channels Used as Single Channel.

Figures 18.29 to 18.33 show connection examples for individual port settings.

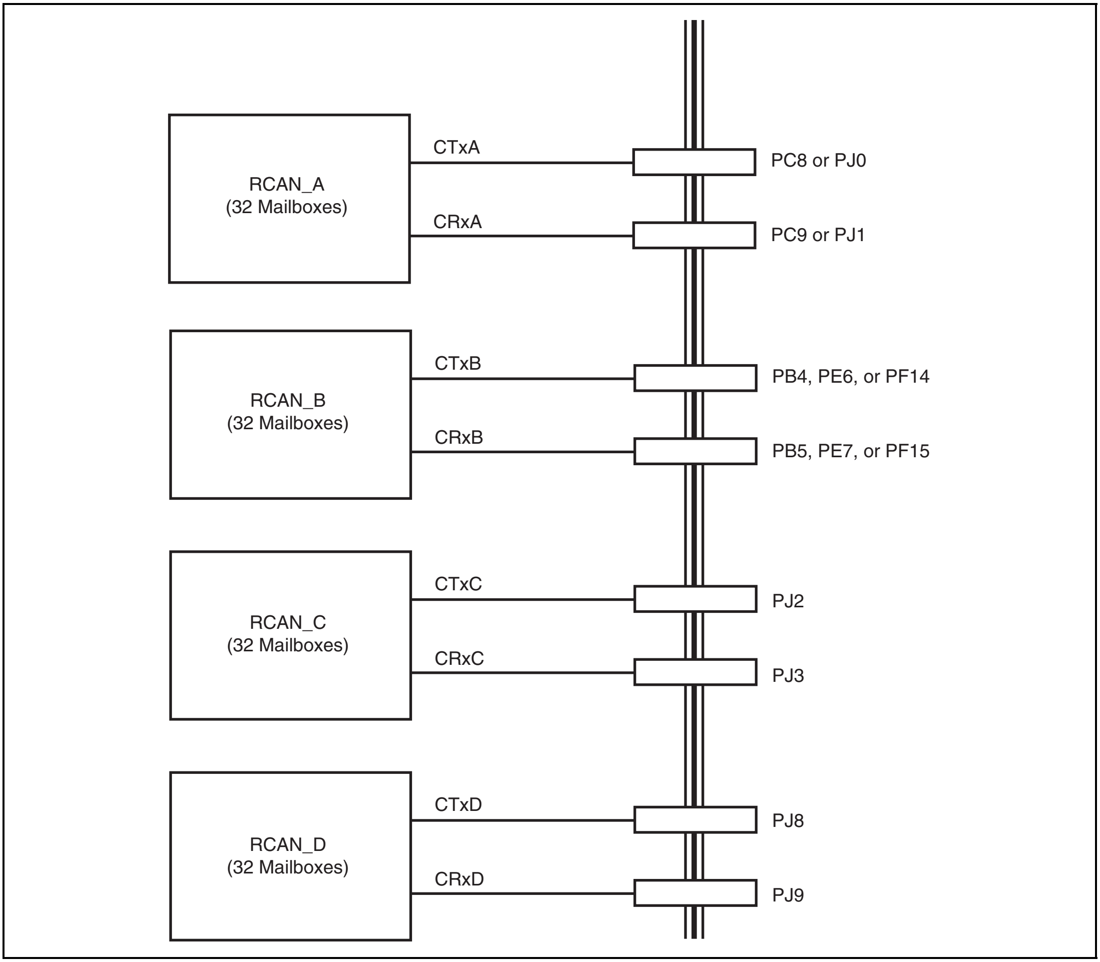


Figure 18.29 Connection Example when Using RCAN as 4-Channel Module (32 Mailboxes × 4 Channels)

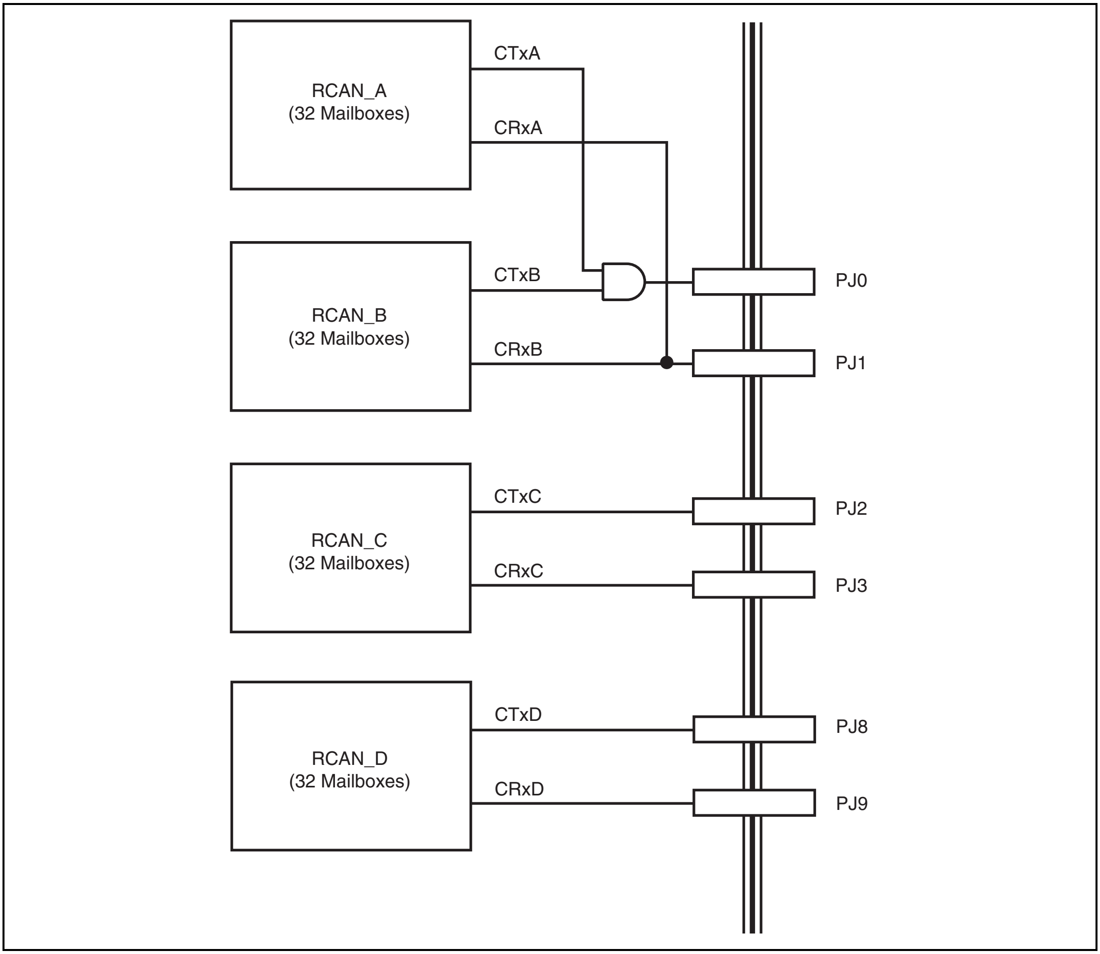


Figure 18.30 Connection Example when Using RCAN as 3-Channel Module (64 Mailboxes × 1 Channel and 32 Mailboxes × 2 Channels)

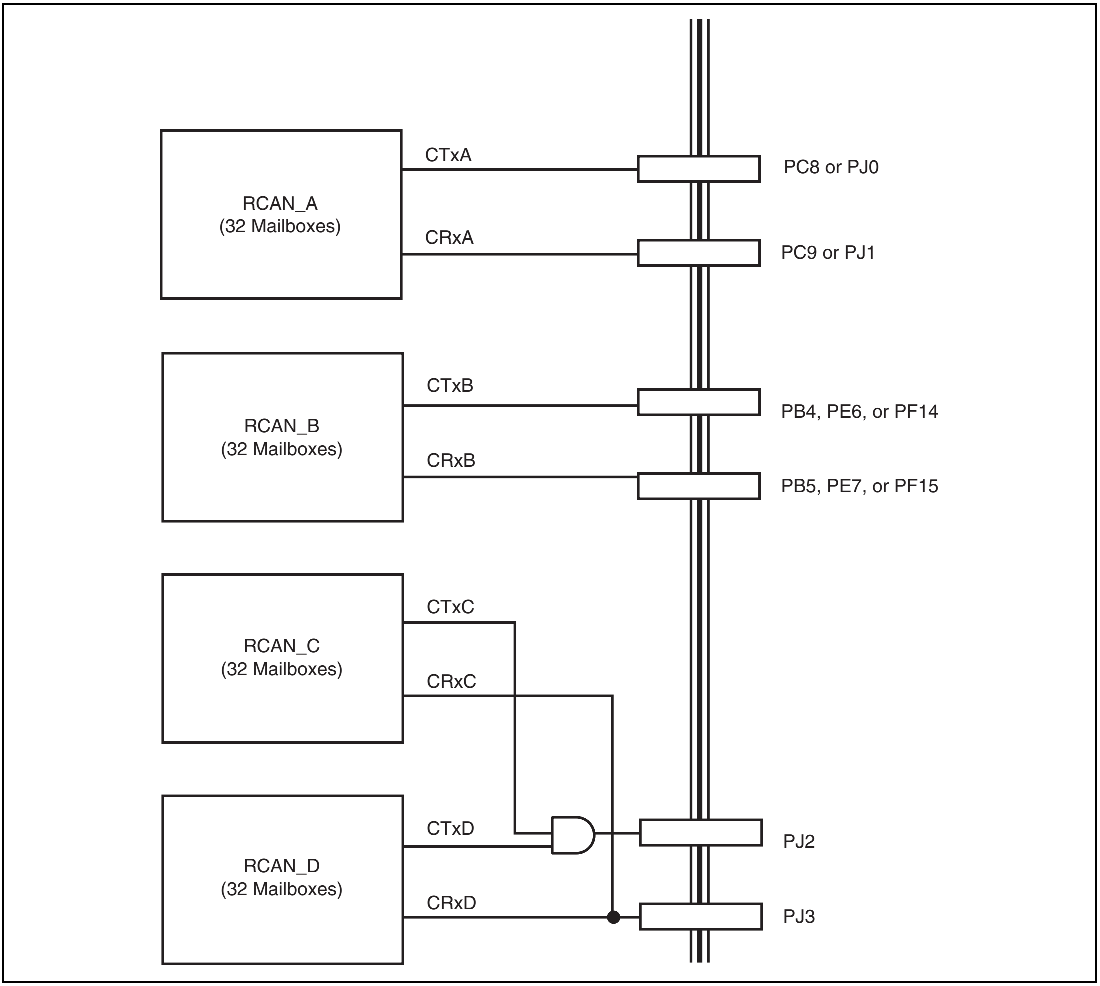


Figure 18.31 Connection Example when Using RCAN as 3-Channel Module (64 Mailboxes × 1 Channel and 32 Mailboxes × 2 Channels)

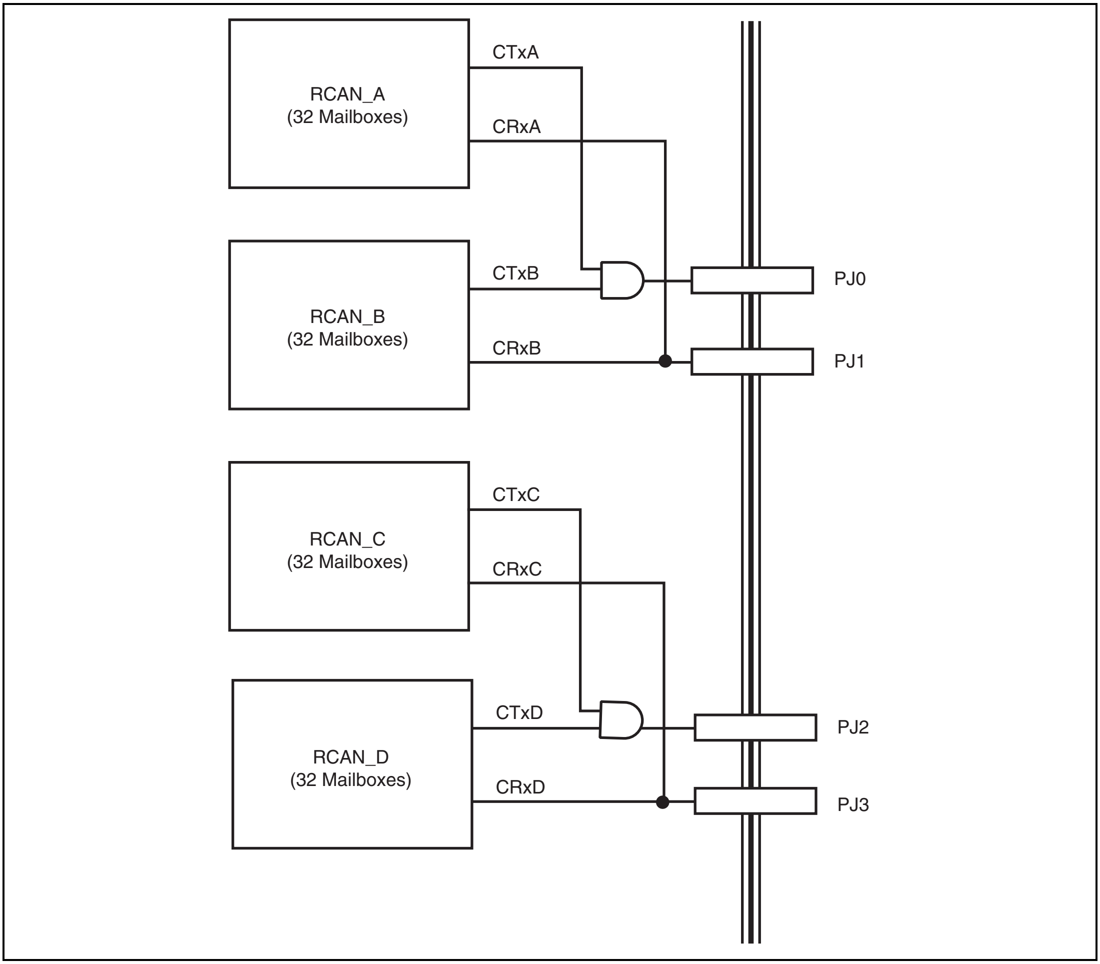


Figure 18.32 Connection Example when Using RCAN as 2-Channel Module (64 Mailboxes × 2 Channels)

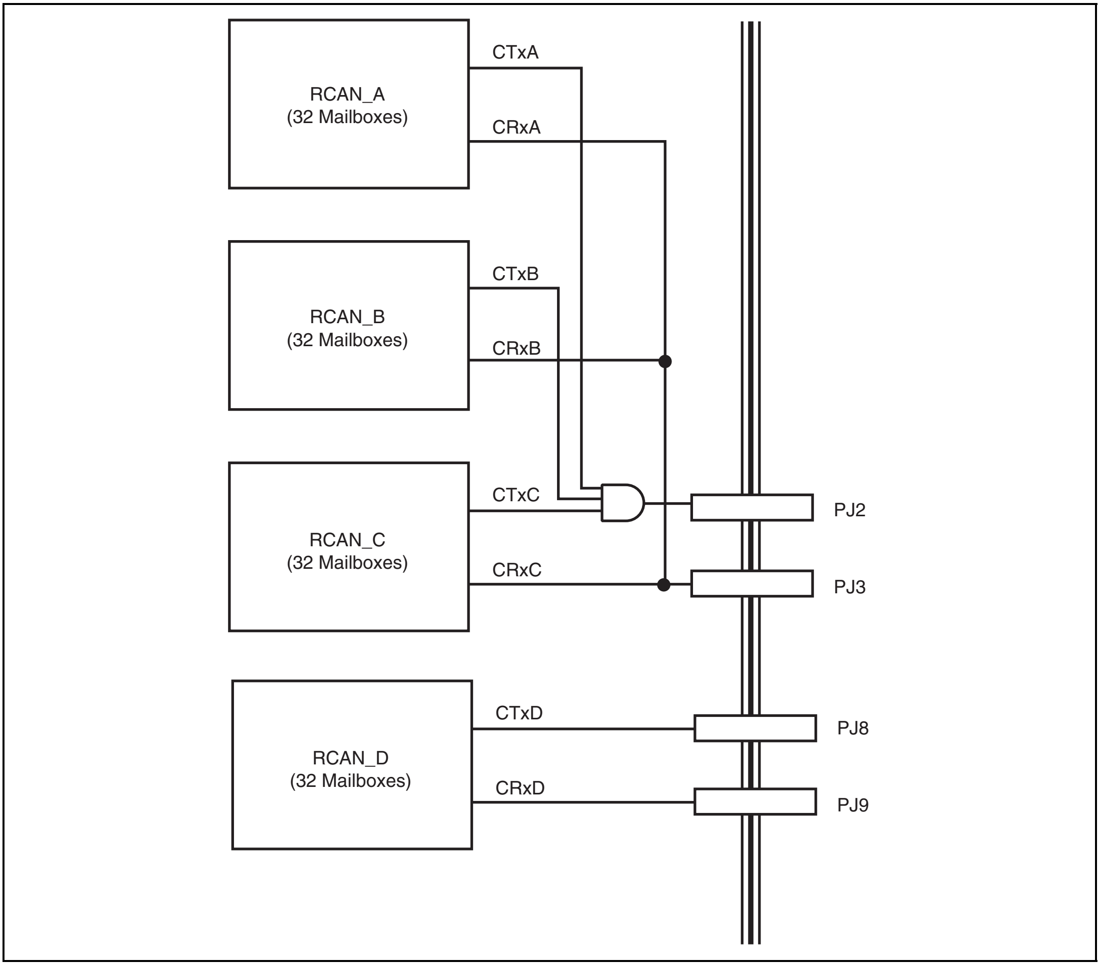


Figure 18.33 Connection Example when Using RCAN as 2-Channel Module (96 Mailboxes × 1 Channel and 32 Mailboxes × 1 Channel)

18.11 Usage Notes

18.11.1 Notes on Port Setting for Multiple Channels Used as Single Channel

The RCAN-TL1 in this LSI has three channels and some of these channels can be used as a single channel. When using multiple channels as a single channel, keep the following in mind.

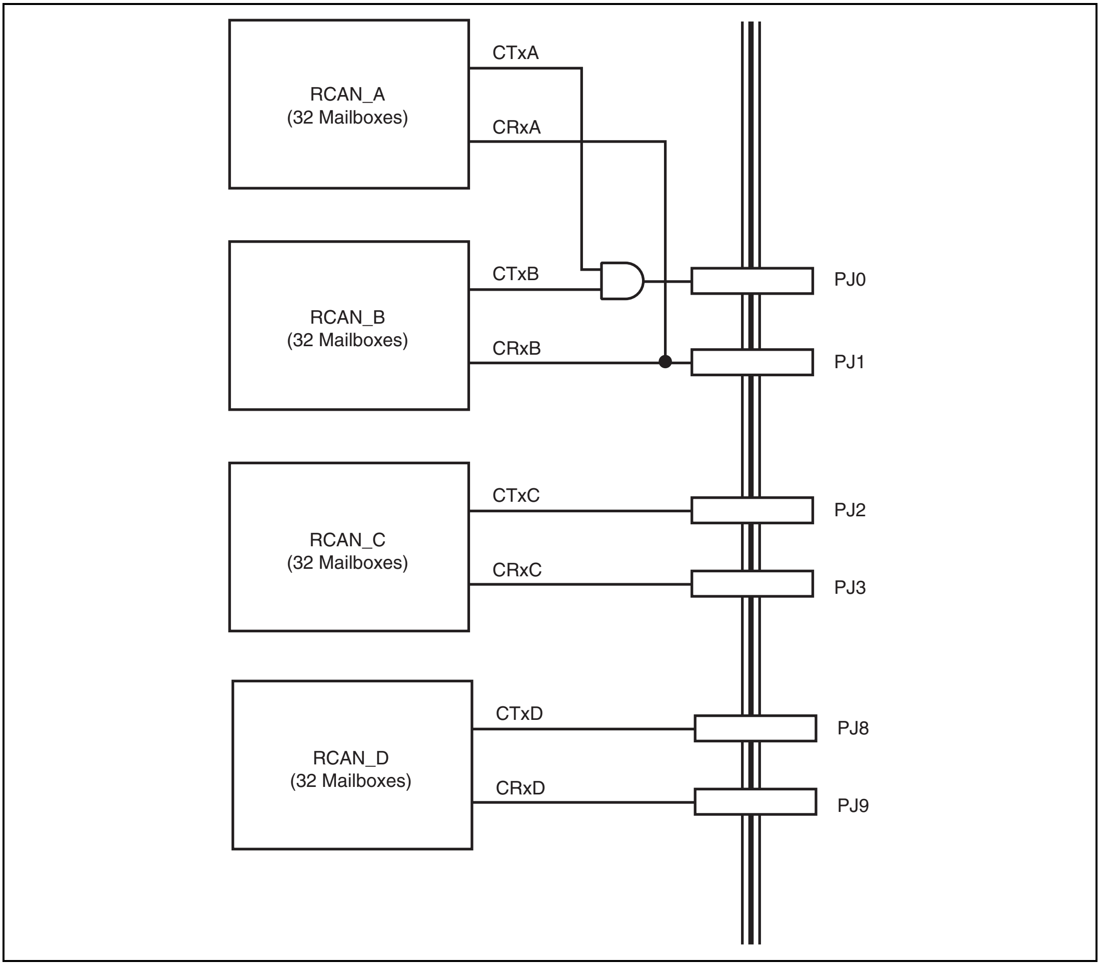


Figure 18.34 Connection Example when Using RCAN as 3-Channel Module (64 Mailboxes × 1 Channel and 32 Mailboxes × 2 Channels)

1. No ACK error is detected even when any other nodes are not connected to the CAN bus. This occurs when channel B transmits an ACK in the ACK field in response to a message channel A has transmitted.

   Channel B receives a message which channel A has transmitted on the CAN bus and then transmits an ACK in the ACK field. After that, channel A receives the ACK.

   To avoid this, make channel B which is not currently used for transmission the listen-only mode (TST[2:0] = B'001) or the reset state (MCR0 = 1). With this setting, only a channel which transmits a message transmits an ACK.

2. Internal arbitration for channels A and B is independently controlled to determine the order of transmission.

   Although the internal arbitration is performed on 31 Mailboxes at a time, it is not performed on 64 Mailboxes at a time even though multiple channels function as a single channel.

3. Do not set the same transmission message ID in both channels A and B.

   Two messages may be transmitted from the two channels after arbitration on the CAN bus.

Section 19 FlexRay Module

The FlexRay Module controls the FlexRay which is an automotive LAN protocol used for the purpose of establishing high-speed communications with high flexibility and reliability in an automobile.

This section describes the program interfaces of the FlexRay Module.

For the data link controller functions of the FlexRay, see the following reference document.

[Reference]

FlexRay Communications System Protocol Specification Version 2.1, 2004-2005 FlexRay Consortium

Note: FlexRay is a registered trademark or trademark of Daimler AG in Japan and other countries.

19.1 Features

- Conformance with FlexRay Communication System Protocol Specification v2.1
- Two channels (channel A and channel B)
- Message RAM: 8 Kbytes

  Up to 30 (payload length: 254 bytes) to 128 (payload length: 48 bytes) transmit/receive buffers configurable
- Receive FIFO: 0 to 128 (shares the Message RAM with transmit/receive buffers)
- Message filtering (following types of filtering can be set at transmit/receive buffers)

  Slot counter filtering

  Channel ID filtering

  Cycle counter filtering
- NM data transmission/reception: Full support for up to 12 bytes

  An interrupt is generated when the NM vector changes
- The following three types of timers can be used (timer value can be set in macrotick precision)

  Timer 0: Absolute value timer

  Timer 1: Relative value timer

  Stop watch timer: Captures cycle counter and MT counter values
- Baud rate: 10 Mbps or 5 Mbps

- Software reset
- Data transfer from the receive message buffer by the A-DMAC

19.2 Terms and Abbreviations

Table 19.1 lists the terms and abbreviations used in this section.

Table 19.1 List of Terms and Abbreviations

Term	Meaning
AP	Action Point
BD	Bus Driver
CAS	Collision Avoidance Symbol
CC	Communication Controller
CHI	Controller Host Interface
CIF	CPU Interface
CRC	Cyclic Redundancy Check
FIFO	First In First Out (message buffer structure)
FSM	Finite State Machine
FSP	Frame and Symbol Processing Block
FTM	Fault Tolerant Midpoint
GTU	Global Time Unit Block
IBF	Input Buffer
INT	Interrupt Control Block
MHD	Message Handler Block
MT	Macrotick
MTS	Media Access Test Symbol
NCT	Network Communication Time
NEM	Network Management Block
NIT	Network Idle Time
NM	Network Management

Term	Meaning
OBF	Output Buffer
POC	Protocol Operation Control
PRT	Protocol Controller Block
SDL	Specification and Description Language
SUC	System Universal Control Block
TBF	Transient Buffer
TDMA	Time Division Multiple Access (media access method)
TSS	Transmission Start Sequence
TT-D	Time Triggered Distributed Synchronization
μT	Microtick
WUP	Wakeup Pattern
WUS	Wakeup Symbol

19.3 Configuration

19.3.1 Block Diagram

The FlexRay Module consists of twelve functional blocks: CPU Interface (CIF), Input Buffer (IBF), Output Buffer (OBF), Message Handler (MHD), Message RAM (MRAM), Transient Buffer (TBF A/B), Protocol Controller (PRT A/B), Global Time Unit (GTU), System Universal Control (SUC), Frame and Symbol Processing (FSP), Network Management (NEM), and Interrupt Control (INT).

Figure 19.1 shows a block diagram of the FlexRay Module.

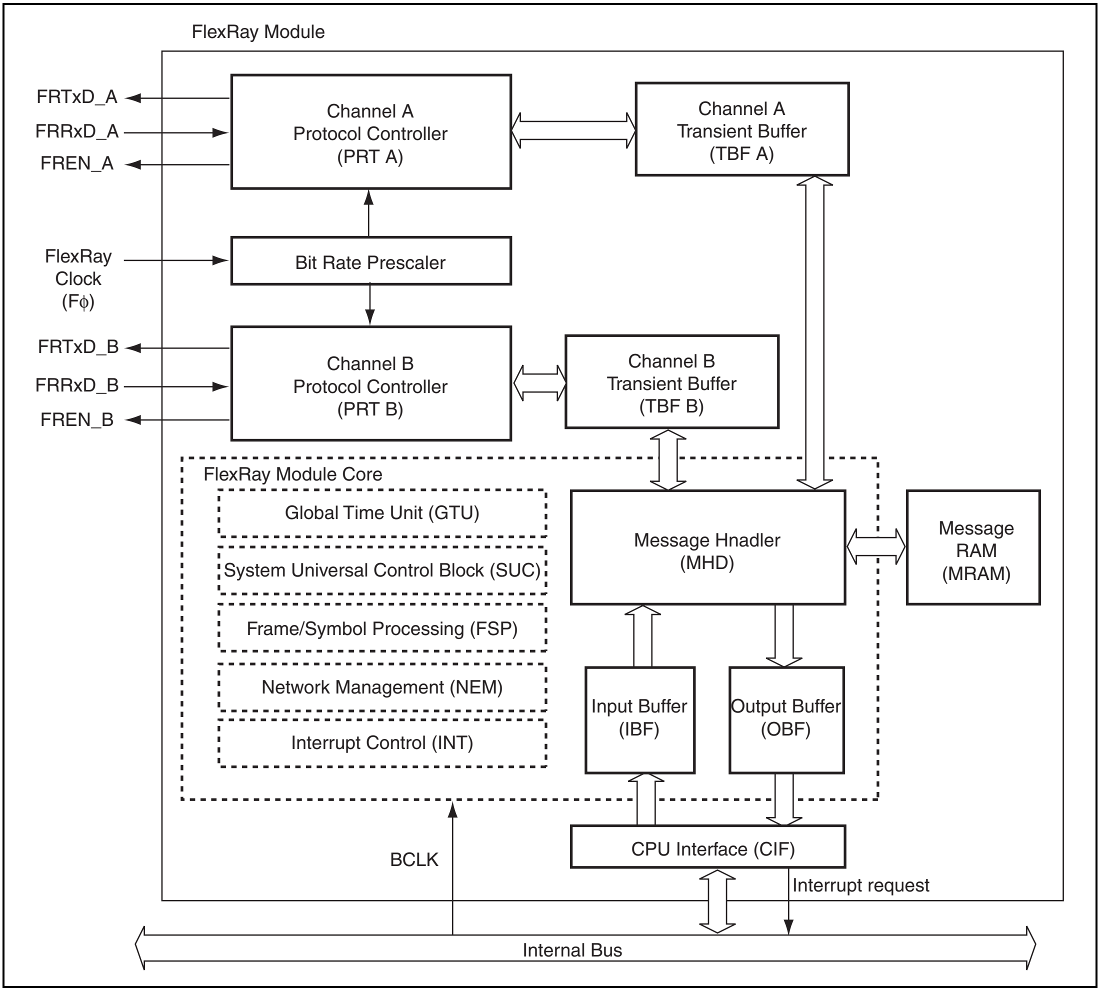


Figure 19.1 Block Diagram of FlexRay Module

19.3.2 Functions of Each Block

(1) CPU Interface (CIF)

Connects the Host CPU to the FlexRay Module via the CPU Interface.

(2) Input Buffer (IBF)

For write access to the message buffers configured in the Message RAM, the Host can write the header and data section for a specific message buffer to the Input Buffer. The Message Handler then transfers the data from the Input Buffer to the selected message buffer in the Message RAM.

(3) Output Buffer (OBF)

For read access to a message buffer configured in the Message RAM the Message Handler transfers the selected message buffer to the Output Buffer. After the transfer has completed, the Host can read the header and data section of the transferred message buffer from the Output Buffer.

(4) Message Handler (MHD)

The FlexRay Message Handler controls data transfers between the following components:

- Input/Output Buffer and Message RAM
- Transient Buffer RAMs of the two FlexRay Protocol Controllers and Message RAM

(5) Message RAM (MRAM)

The Message RAM consists of a single-ported RAM that stores up to 128 FlexRay message buffers together with the related configuration data (header and data partition).

(6) Transient Buffer (TBF A/B)

Stores the data section of two complete messages.

(7) Protocol Controller (PRT A/B)

The FlexRay Channel Protocol Controllers consist of shift register and FlexRay protocol FSM. They are connected to the Transient Buffer RAMs for intermediate message storage and to the physical layer via bus driver BD.

They perform the following functionality:

- Control and check of bit timing
- Reception/transmission of FlexRay frames and symbols
- Check of header CRC
- Generation/check of frame CRC
- Interfacing to bus driver

The FlexRay Channel Protocol Controllers have interfaces to:

- Physical Layer (bus driver)
- Transient Buffer RAM
- Message Handler
- Global Time Unit
- System Universal Control
- Frame and Symbol Processing
- Network Management
- Interrupt Control

(8) Global Time Unit (GTU)

The Global Time Unit performs the following functions:

- Generation of microtick
- Generation of macrotick
- Fault tolerant clock synchronization by FTM algorithm
  - Rate correction
  - Offset correction
- Cycle counter
- Timing control of static segment
- Timing control of dynamic segment (minislotting)
- Support of external clock correction

(9) System Universal Control Block (SUC)

The System Universal Control controls the following functions:

- Configuration
- Wakeup
- Startup
- Normal Operation
- Passive Operation

(10) Frame and Symbol Processing (FSP)

The Frame and Symbol Processing controls the following functions:

- Checks the correct timing of frames and symbols
- Tests the syntactical and semantical correctness of received frames
- Sets the slot status flags

(11) Network Management (NEM)

Handles the FlexRay network management vector.

(12) Interrupt Control (INT)

The Interrupt Controller performs the following functions:

- Provides error and status interrupt flags
- Enables/disables interrupt sources
- Assignment of interrupt sources to one of the two module interrupt lines
- Enables/disables module interrupt lines
- Manages the two interrupt timers
- Stop watch time capturing

19.4 Pin Configuration

Table 19.2 shows the pin configuration of the FlexRay Module.

Table 19.2 Pin Configuration

Pin Name	Symbol	I/O	Function
Channel A transmit data pin	FRTxD_A	Output	Transmit data output pin for FlexRay bus channel A
Channel A receive data pin	FRRxD_A	Input	Receive data output pin for FlexRay bus channel A
Channel A transmit data enable pin	FREN_A	Output	Transmit data enable pin for FlexRay bus channel A High: Transmission disabled Low: Transmission enabled
Channel B transmit data pin	FRTxD_B	Output	Transmit data output pin for FlexRay bus channel B
Channel B receive data pin	FRRxD_B	Input	Receive data output pin for FlexRay bus channel B
Channel B transmit data enable pin	FREN_B	Output	Transmit data enable pin for FlexRay bus channel B High: Transmission disabled Low: Transmission enabled

19.5 Register Map

	Address	Symbol	Name	Reset	Access	Block	
Special	Special Registers						
	H'FFFC2004	FXROC	FlexRay Operation Control Register	04	r/w	CIF	Customer I/F Block
	H'FFFC2005	FXROS	FlexRay Operation Status Register	00	r		
	H'FFFC200C	FXRTISR	FlexRay Timer Interrupt Status Register	00	r/w		
	H'FFFC200D	FXRTIER	FlexRay Timer Interrupt Enable Register	00	r/w		
	H'FFFC201C	FRLCK	FlexRay Lock Register	0000 0000	r/w	GIF	General I/F Block
Interrupt	Interrupt Registers						
	H'FFFC2020	FREIR	FlexRay Error Interrupt Register	0000 0000	r/w	INT	Interrupt Control Block
	H'FFFC2024	FRSIR	FlexRay Status Interrupt Register	0000 0000	r/w		
	H'FFFC2028	FREILS	FlexRay Error Interrupt Line Select	0000 0000	r/w		
	H'FFFC202C	FRSILS	FlexRay Status Interrupt Line Select	0303 FFFF	r/w		
	H'FFFC2030	FREIES	FlexRay Error Interrupt Enable Set	0000 0000	r/w		
	H'FFFC2034	FREIER	FlexRay Error Interrupt Enable Reset	0000 0000	r/w		
	H'FFFC2038	FRSIES	FlexRay Status Interrupt Enable Set	0000 0000	r/w		
	H'FFFC203C	FRSIER	FlexRay Status Interrupt Enable Reset	0000 0000	r/w		
	H'FFFC2040	FRILE	FlexRay Interrupt Line Enable	0000 0000	r/w		
	H'FFFC2044	FRT0C	FlexRay Timer 0 Configuration	0000 0000	r/w		
	H'FFFC2048	FRT1C	FlexRay Timer 1 Configuration	0002 0000	r/w		
	H'FFFC204C	FRSTPW1	FlexRay Stop Watch Register 1	0000 0000	r/w		
	H'FFFC2050	FRSTPW2	FlexRay Stop Watch Register 2	0000 0000	r/w		
Communication Control	CC Control Registers (CC: Communication Controller)						
	H'FFFC2080	FRSUCC1	FlexRay SUC Configuration Register 1	0C40 1080	r/w	SUC	System Universal Control Block
	H'FFFC2084	FRSUCC2	FlexRay SUC Configuration Register 2	0100 0504	r/w		
	H'FFFC2088	FRSUCC3	FlexRay SUC Configuration Register 3	0000 0011	r/w		
	H'FFFC208C	FRNEMC	FlexRay NEM Configuration Register	0000 0000	r/w	NEM	Network Management Block
	H'FFFC2090	FRPRTC1	FlexRay PRT Configuration Register 1	084C 0633	r/w	PRT	Protocol Controller Block
	H'FFFC2094	FRPRTC2	FlexRay PRT Configuration Register 2	0F2D 0A0E	r/w		

	Address	Symbol	Name	Reset	Access	Block	
Commu-nication Control	H'FFFC2098	FRMHDC	FlexRay MHD Configuration Register	0000 0000	r/w	MHD	Message Handler Block
	H'FFFC20A0	FRGTUC1	FlexRay GTU Configuration Register 1	0000 0280	r/w	GTU	Global Time Unit Block
	H'FFFC20A4	FRGTUC2	FlexRay GTU Configuration Register 2	0002 000A	r/w		
	H'FFFC20A8	FRGTUC3	FlexRay GTU Configuration Register 3	0202 0000	r/w		
	H'FFFC20AC	FRGTUC4	FlexRay GTU Configuration Register 4	0008 0007	r/w		
	H'FFFC20B0	FRGTUC5	FlexRay GTU Configuration Register 5	0E00 0000	r/w		
	H'FFFC20B4	FRGTUC6	FlexRay GTU Configuration Register 6	0002 0000	r/w		
	H'FFFC20B8	FRGTUC7	FlexRay GTU Configuration Register 7	0002 0004	r/w		
	H'FFFC20BC	FRGTUC8	FlexRay GTU Configuration Register 8	0000 0002	r/w		
	H'FFFC20C0	FRGTUC9	FlexRay GTU Configuration Register 9	0000 0101	r/w		
	H'FFFC20C4	FRGTUC10	FlexRay GTU Configuration Register 10	0002 0005	r/w		
	H'FFFC20C8	FRGTUC11	FlexRay GTU Configuration Register 11	0000 0000	r/w		
Commu-nication Status	CC Status Registers						
	H'FFFC2100	FRCCSV	FlexRay CC Status Vector	0010 4000	r	SUC	System Universal Control Block
	H'FFFC2104	FRCCEV	FlexRay CC Error Vector	0000 0000	r		
	H'FFFC2110	FRSCV	FlexRay Slot Counter Value	0000 0000	r	GTU	Global Time Unit Block
	H'FFFC2114	FRMTCCV	FlexRay Macrotick and Cycle Counter Value	0000 0000	r		
	H'FFFC2118	FRRCV	FlexRay Rate Correction Value	0000 0000	r		
	H'FFFC211C	FROCV	FlexRay Offset Correction Value	0000 0000	r		
	H'FFFC2120	FRSFS	FlexRay Sync Frame Status	0000 0000	r		
	H'FFFC2124	FRSWNIT	FlexRay Symbol Window and NIT Status	0000 0000	r		
	H'FFFC2128	FRACS	FlexRay Aggregated Channel Status	0000 0000	r/w		
	H'FFFC2130 to H'FFFC2168	FRESIDn	FlexRay Even Sync ID n (n = 1 to 15)	0000 0000	r		
	H'FFFC2170 to H'FFFC21A8	FROSIDn	FlexRay Odd Sync ID n (n = 1 to 15)	0000 0000	r		
	H'FFFC21B0 to H'FFFC21B8	FRNMVn	FlexRay Network Management Vector n (n = 1 to 3)	0000 0000	r	NEM	Network Manage-ment Block
Message Buffer Control	Message Buffer Control Registers						
	H'FFFC2300	FRMRC	FlexRay Message RAM Configuration	0180 0000	r/w	MHD	Message Handler Block
	H'FFFC2304	FRFRF	FlexRay FIFO Rejection Filter	0180 0000	r/w		
	H'FFFC2308	FRFRFM	FlexRay FIFO Rejection Filter Mask	0000 0000	r/w		
	H'FFFC230C	FRFCL	FlexRay FIFO Critical Level	0000 0080	r/w		

	Address	Symbol	Name	Reset	Access	Block	
Message Buffer Control	H'FFFC2310	FRMHDS	FlexRay Message Handler Status	0000 0080	r/w	MHD	Message Handler Block
	H'FFFC2314	FRLDTS	FlexRay Last Dynamic Transmit Slot	0000 0000	r		
	H'FFFC2318	FRFSR	FlexRay FIFO Status Register	0000 0000	r		
	H'FFFC231C	FRMHDF	FlexRay Message Handler Constraints Flags	0000 0000	r/w		
	H'FFFC2320	FRTXRQ1	FlexRay Transmission Request 1	0000 0000	r		
	H'FFFC2324	FRTXRQ2	FlexRay Transmission Request 2	0000 0000	r		
	H'FFFC2328	FRTXRQ3	FlexRay Transmission Request 3	0000 0000	r		
	H'FFFC232C	FRTXRQ4	FlexRay Transmission Request 4	0000 0000	r		
	H'FFFC2330	FRNDAT1	FlexRay New Data 1	0000 0000	r		
	H'FFFC2334	FRNDAT2	FlexRay New Data 2	0000 0000	r		
	H'FFFC2338	FRNDAT3	FlexRay New Data 3	0000 0000	r		
	H'FFFC233C	FRNDAT4	FlexRay New Data 4	0000 0000	r		
	H'FFFC2340	FRMBSC1	FlexRay Message Buffer Status Changed 1	0000 0000	r		
	H'FFFC2344	FRMBSC2	FlexRay Message Buffer Status Changed 2	0000 0000	r		
	H'FFFC2348	FRMBSC3	FlexRay Message Buffer Status Changed 3	0000 0000	r		
	H'FFFC234C	FRMBSC4	FlexRay Message Buffer Status Changed 4	0000 0000	r		
Input Buffer	Input Buffer						
	H'FFFC2400 to H'FFFC24FC	FRWRDSn	FlexRay Write Data Section n (n = 1 to 64)	0000 0000	r/w	IBF	Input Buffer
	H'FFFC2500	FRWRHS1	FlexRay Write Header Section 1	0000 0000	r/w		
	H'FFFC2504	FRWRHS2	FlexRay Write Header Section 2	0000 0000	r/w		
	H'FFFC2508	FRWRHS3	FlexRay Write Header Section 3	0000 0000	r/w		
	H'FFFC2510	FRIBCM	FlexRay Input Buffer Command Mask	0000 0000	r/w		
	H'FFFC2514	FRIBCR	FlexRay Input Buffer Command Request	0000 0000	r/w		
Output Buffer	Output Buffer						
	H'FFFC2600 to H'FFFC26FC	FRRDDSn	FlexRay Read Data Section n (n = 1 to 64)	0000 0000	r	OBF	Output Buffer
	H'FFFC2700	FRRDHS1	FlexRay Read Header Section 1	0000 0000	r		
	H'FFFC2704	FRRDHS2	FlexRay Read Header Section 2	0000 0000	r		
	H'FFFC2708	FRRDHS3	FlexRay Read Header Section 3	0000 0000	r		
	H'FFFC270C	FRMBS	FlexRay Message Buffer Status	0000 0000	r		
	H'FFFC2710	FROBCM	FlexRay Output Buffer Command Mask	0000 0000	r/w		
	H'FFFC2714	FROBCR	FlexRay Output Buffer Command Request	0000 0000	r/w		

19.6 Special Registers

19.6.1 FlexRay Operation Control Register (FXROC)

FXROC is an 8-bit readable/writable register that controls the FlexRay operation.

The CPU can always read from or write to FXROC.

FXROC is initialized to H'04 by a power-on reset or in hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	FOPC	—	—	—	—	FEC	—	FOPEN
Initial value:	0	0	0	0	0	1	0	0
R/W:	R/W	R	R	R	R	R/W	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	FOPC	0	R/W	FlexRay Operation Control Protect Enables or disables write access to the FOPEN bit. 0: Write access to FOPEN enabled 1: Write access to FOPEN disabled
6 to 3	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
2	FEC	1	R/W	FlexRay Endian Control Enables or disables the byte swap for read or write access to the FlexRay network management vector [1...3] (FRNMVn), FlexRay write data section [1...64] (FRWRDSn), and FlexRay read data section [1...64] (FRRDDSn). 0: Byte swap disabled 1: Byte swap enabled For the byte order control, refer to Note below.
1	—	0	R	Reserved. This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
0	FOPEN	0	R/W	FlexRay Enable Enables or disables the FlexRay operation. This bit can be accessed only when the FOPC bit is "0". 0: Operation disabled When this bit is cleared to "0", all FlexRay registers except FXROC are initialized and read and write access is disabled regardless of the FlexRay operation status. Transition from the operation-enabled state to operation-disabled state takes three peripheral clock cycles. After specifying transition from the operation-disabled state to operation-enabled state, do not disable the FlexRay operation while the FRSTAT bit indicates the reset state. 1: Operation enabled When this bit is set to "1", read and write access to all FlexRay registers is enabled. Immediately after this bit is set to 1, the registers except FXROC hold the default values. Before starting communication through FlexRay, be sure to set this bit to "1". Transition from the operation-disabled state to operation-enabled state takes three peripheral clock cycles. After specifying transition from the operation-enabled state to operation-disabled state, do not enable the FlexRay operation while the FRSTAT bit indicates the operating state.

Note:

Byte Swap Disabled:

FRNMVn

	31 to 24	23 to 16	15 to 8	7 to 0
FRNMV1	Data3	Data2	Data1	Data0
FRNMV2	Data7	Data6	Data5	Data4
FRNMV3	Data11	Data10	Data9	Data8

FRWRDSn

	31 to 24	23 to 16	15 to 8	7 to 0
FRWRDSn	DW_{2n}, $byte_{4n-1}$	DW_{2n}, $byte_{4n-2}$	DW_{2n-1}, $byte_{4n-3}$	DW_{2n-1}, $byte_{4n-4}$

FRRDDSn

	31 to 24	23 to 16	15 to 8	7 to 0
FRRDDSn	DW_{2n}, $byte_{4n-1}$	DW_{2n}, $byte_{4n-2}$	DW_{2n-1}, $byte_{4n-3}$	DW_{2n-1}, $byte_{4n-4}$

Byte Swap Enabled:

FRNMVn

	31 to 24	23 to 16	15 to 8	7 to 0
FRNMV1	Data0	Data1	Data2	Data3
FRNMV2	Data4	Data5	Data6	Data7
FRNMV3	Data8	Data9	Data10	Data11

FRWRDSn

	31 to 24	23 to 16	15 to 8	7 to 0
FRWRDSn	DW_{2n-1}, $byte_{4n-4}$	DW_{2n-1}, $byte_{4n-3}$	DW_{2n}, $byte_{4n-2}$	DW_{2n}, $byte_{4n-1}$

FRRDDSn

	31 to 24	23 to 16	15 to 8	7 to 0
FRRDDSn	DW_{2n-1}, $byte_{4n-4}$	DW_{2n-1}, $byte_{4n-3}$	DW_{2n}, $byte_{4n-2}$	DW_{2n}, $byte_{4n-1}$

19.6.2 FlexRay Operation Status Register (FXROS)

FXROS is an 8-bit read-only register that indicates the FlexRay operation status.

The CPU can always read FXROS. Note that FRSTAT is a read-only flag and cannot be written to.

FXROS is initialized to H'00 by a power-on reset or in hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	FRSTAT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
0	FRSTAT	0	R	FlexRay Reset Status Indicates the FlexRay operation status. Writing to this bit is ignored. 0: Reset state In this state, only FXROC can be accessed. The other registers are initialized and cannot be accessed. 1: Operating state In this state, all FlexRay registers can be accessed.

19.6.3 FlexRay Timer Interrupt Status Register (FXRTISR)

FXRTISR is an 8-bit readable/writable register that indicates the FlexRay timer interrupt status.

The FT1IS and FT0IS flags in FXRTISR cannot be set by writing "1" to them; "1" can be written only to clear them.

FXRTISR is initialized to H'00 by a power-on reset or in hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	FT1IS	FT0IS
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/(W)*	R/(W)*

Note: * Only 1 can be written to clear the flag.

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
1	FT1IS	0	R/W	FlexRay Timer 1 Interrupt Status Indicates whether a timer 1 interrupt has occurred. 0: No timer 1 interrupt has occurred. [Clearing condition] • 1 is written while FT1IS = 1. 1: A timer 1 interrupt has occurred. [Setting condition] • The condition specified in the FlexRay timer 1 configuration (FRT1C) is satisfied. Note: This flag is set regardless of the setting in the FlexRay timer interrupt enable register (FXRTIER).

Bit	Bit Name	Initial Value	R/W	Description
0	FT0IS	0	R/W	FlexRay Timer 0 Interrupt Status Indicates whether a timer 0 interrupt has occurred. 0: No timer 0 interrupt has occurred. [Clearing condition] • 1 is written while FT0IS = 1. 1: A timer 0 interrupt has occurred. [Setting condition] • The condition specified in the FlexRay timer 0 configuration (FRT0C) is satisfied. Note: This flag is set regardless of the setting in the FlexRay timer interrupt enable register (FXRTIER).

19.6.4 FlexRay Timer Interrupt Enable Register (FXRTIER)

FXRTIER is an 8-bit readable/writable register that enables or disables occurrence of timer 0 and 1 interrupt requests in the FlexRay module.

The CPU can always read from or write to FXRTIER.

FXRTIER is initialized to H'00 by a power-on reset or in hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	FT1IEN	FT0IEN
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
1	FT1IEN	0	R/W	FlexRay Timer 1 Interrupt Enable Enables or disables occurrence of FlexRay timer 1 interrupt requests. 0: Timer 1 interrupt requests disabled 1: Timer 1 interrupt requests enabled
0	FT0IEN	0	R/W	FlexRay Timer 0 Interrupt Enable Enables or disables occurrence of FlexRay timer 0 interrupt requests. 0: Timer 0 interrupt requests disabled 1: Timer 0 interrupt requests enabled

19.6.5 FlexRay Lock Register (FRLCK)

The FlexRay lock register is write-only. Reading the register will return 0x0000 0000.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	FRLCK							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
7 to 0	FRLCK	All 0	W	Configuration Lock Key To leave CONFIG state by writing FRSUCC1.CMD[3:0] (command READY), the write operation has to be directly preceded by two write accesses to the Configuration Lock Key (unlock sequence). If the write sequence below is interrupted by other write accesses between the second write to the Configuration Lock Key and the write access to the FRSUCC1 register, the CC remains in CONFIG state and the sequence has to be repeated. First write: FRLCK.CLK[7:0] = "1100 1110" (0xCE) Second write: FRLCK.CLK[7:0] = "0011 0001" (0x31) Third write: FRSUCC1.CMD[3:0]

Note: In case that the Host uses 8-/16-bit accesses to write the listed bit fields, the programmer has to ensure that no "dummy accesses" e.g. to the remaining register bytes/words are inserted by the compiler.

19.7 Interrupt Registers

19.7.1 FlexRay Error Interrupt Register (FREIR)

The flags are set when the CC detects one of the listed events. The flags remain set until the Host clears them. A flag is cleared by writing a '1' to the corresponding bit position. Writing a '0' has no effect on the flag. A hardware reset or a software reset will also clear the register.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	TABB	LTVB	EDB	—	—	—	—	—	TABA	LTVA	EDA
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/(W)*	R/(W)*	R/(W)*	R	R	R	R	R	R/(W)*	R/(W)*	R/(W)*

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	MHF	IOBA	IIBA	EFA	RFO	PERR	CCL	CCF	SFO	SFBM	CNA	PEMC
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only 1 can be written to clear the flag.

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
26	TABB	0	R/W	Transmission Across Boundary Channel B The flag signals to the Host that a transmission across a slot boundary occurred for channel B. 0: No transmission across slot boundary detected on channel B 1: Transmission across slot boundary detected on channel B
25	LTVB	0	R/W	Latest Transmit Violation Channel B The flag signals a latest transmit violation on channel B to the Host. 0: No latest transmit violation detected on channel B 1: Latest transmit violation detected on channel B

Bit	Bit Name	Initial Value	R/W	Description
24	EDB	0	R/W	Error Detected on Channel B This bit is set whenever one of the flags FRACS.SEDB, FRACS.CEDB, FRACS.CIB, and FRACS.SBVB changes from '0' to '1'. 0: No error detected on channel B 1: Error detected on channel B
23 to 19	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
18	TABA	0	R/W	Transmission Across Boundary Channel A The flag signals to the Host that a transmission across a slot boundary occurred for channel A. 0: No transmission across slot boundary detected on channel A 1: Transmission across slot boundary detected on channel A
17	LTVA	0	R/W	Latest Transmit Violation Channel A The flag signals a latest transmit violation on channel A to the Host. 0: No latest transmit violation detected on channel A 1: Latest transmit violation detected on channel A
16	EDA	0	R/W	Error Detected on Channel A This bit is set whenever one of the flags FRACS.SEDA, FRACS.CEDA, FRACS.CIA, and FRACS.SBVA changes from '0' to '1'. 0: No error detected on channel A 1: Error detected on channel A
15 to 12	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
11	MHF	0	R/W	Message Handler Constraints Flag The flag signals a Message Handler constraints violation condition. It is set whenever one of the flags FRMHDF.SNUA, FRMHDF.SNUB, FRMHDF.FNFA, FRMHDF.FNFB, FRMHDF.TBFA, FRMHDF.TBFB, and FRMHDF.WAHP changes from '0' to '1'. 0: No Message Handler failure detected 1: Message Handler failure detected
10	IOBA	0	R/W	Illegal Output Buffer Access This flag is set by the CC when the Host requests the transfer of a message buffer from the Message RAM to the Output Buffer while FROBCR.OBSYS is set to '1'. 0: No illegal Host access to Output Buffer occurred 1: Illegal Host access to Output Buffer occurred
9	IIBA	0	R/W	Illegal Input Buffer Access This flag is set by the CC when the Host wants to modify a message buffer via Input Buffer and one of the following conditions applies: 1) The CC is not in CONFIG or DEFAULT_CONFIG state and the Host writes to the Input Buffer Command Request register to modify the — Header section of message buffer 0, 1 if configured for transmission in key slot — Header section of static message buffers with buffer number < FRMRC.FDB[7:0] while FRMRC.SEC[1:0] = "01" — Header section of any static or dynamic message buffer while FRMRC.SEC[1:0] = "1x" — Header and/or data section of any message buffer belonging to the receive FIFO 2) The Host writes to any register of the Input Buffer while FRIBCR.IBSYH is set to '1'. 0: No illegal Host access to Input Buffer occurred 1: Illegal Host access to Input Buffer occurred

Bit	Bit Name	Initial Value	R/W	Description
8	EFA	0	R/W	Empty FIFO Access This flag is set by the CC when the Host requests the transfer of a message from the receive FIFO via Output Buffer while the receive FIFO is empty. 0: No Host access to empty FIFO occurred 1: Host access to empty FIFO occurred
7	RFO	0	R/W	Receive FIFO Overrun The flag is set by the CC when a receive FIFO overrun is detected. When a receive FIFO overrun occurs, the oldest message is overwritten with the actual received message. The actual state of the FIFO is monitored in register FRFSR. 0: No receive FIFO overrun detected 1: A receive FIFO overrun has been detected
6	PERR	0	R/W	Parity Error The flag signals a parity error to the Host. It is set whenever one of the flags FRMHDS.PIBF, FRMHDS.POBF, FRMHDS.PMR, FRMHDS.PTBF1, and FRMHDS.PTBF2 changes from '0' to '1'. 0: No parity error detected 1: Parity error detected
5	CCL	0	R/W	CHI Command Locked The flag signals that the write access to the CHI command vector FRSUCC1.CMD[3:0] was not successful because the execution of the previous CHI command has not yet completed. In this case bit CNA is also set to '1'. 0: CHI command accepted 1: CHI command not accepted

Bit	Bit Name	Initial Value	R/W	Description
4	CCF	0	R/W	Clock Correction Failure This flag is set at the end of the cycle whenever one of the following errors occurred: • Missing offset and/or rate correction • Clock correction limit reached The clock correction status is monitored in registers FRCCEV and FRSFS. A failure may occur during startup, therefore bit CCF should be cleared by the Host after the CC entered NORMAL_ACTIVE state. 0: No clock correction error 1: Clock correction failed
3	SFO	0	R/W	Sync Frame Overflow Set when either the number of sync frames received during the last communication cycle or the total number of different sync frame IDs received during the last double cycle exceeds the maximum number of sync frames as defined by FRGTUC2.SNM[3:0]. 0: Number of received sync frames ≤ FRGTUC2.SNM[3:0] 1: More sync frames received than configured by FRGTUC2.SNM[3:0]
2	SFBM	0	R/W	Sync Frames Below Minimum This flag signals that the number of sync frames received during the last communication cycle was below the limit required by the FlexRay protocol. May be set during startup and therefore should be cleared by the Host after the CC entered NORMAL_ACTIVE state. 0: Sync node: 1 or more sync frames received, non-sync node: 2 or more sync frames received 1: Less than the required minimum of sync frames received

Bit	Bit Name	Initial Value	R/W	Description
1	CNA	0	R/W	Command Not Accepted The flag signals that the write access to the CHI command vector FRSUCC1.CMD[3:0] was not successful because the requested command was not valid in the actual POC state, or because the CHI command was locked (CCL = '1'). 0: CHI command accepted 1: CHI command not accepted
0	PEMC	0	R/W	POC Error Mode Changed This flag is set whenever the error mode signalled by FRCCEV.ERRM[1:0] has changed. 0: Error mode has not changed 1: Error mode has changed

19.7.2 FlexRay Status Interrupt Register (FRSIR)

The flags are set when the CC detects one of the listed events. The flags remain set until the Host clears them. A flag is cleared by writing a '1' to the corresponding bit position. Writing a '0' has no effect on the flag. A hardware reset or a software reset will also clear the register.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	MTSB	WUPB	—	—	—	—	—	—	MTSA	WUPA
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/(W)*	R/(W)*	R	R	R	R	R	R	R/(W)*	R/(W)*

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SDS	MBSI	SUCS	SWE	TOBC	TIBC	TI1	TI0	NMVC	RFCL	RFNE	RXI	TXI	CYCS	CAS	WST
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only 1 can be written to clear the flag.

Bit	Bit Name	Initial Value	R/W	Description
31 to 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
25	MTSB	0	R/W	MTS Received on Channel B (vSS!ValidMTSB) Media Access Test symbol received on channel B during the preceding symbol window. Updated by the CC for each channel at the end of the symbol window. 0: No MTS symbol received on channel B 1: MTS symbol received on channel B
24	WUPB	0	R/W	Wakeup Pattern Channel B This flag is set by the CC when a wakeup pattern was received on channel B. Only set when the CC is in WAKEUP, READY, or STARTUP state. 0: No wakeup pattern received on channel B 1: Wakeup pattern received on channel B
23 to 18	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
17	MTSA	0	R/W	MTS Received on Channel A (vSS!ValidMTSA) Media Access Test symbol received on channel A during the preceding symbol window. Updated by the CC for each channel at the end of the symbol window. 0: No MTS symbol received on channel A 1: MTS symbol received on channel A
16	WUPA	0	R/W	Wakeup Pattern Channel A This flag is set by the CC when a wakeup pattern was received on channel A. Only set when the CC is in WAKEUP, READY, or STARTUP state. 0: No wakeup pattern received on channel A 1: Wakeup pattern received on channel A
15	SDS	0	R/W	Start of Dynamic Segment This flag is set by the CC when the dynamic segment starts. 0: Dynamic segment not yet started 1: Dynamic segment started
14	MBSI	0	R/W	Message Buffer Status Interrupt This flag is set by the CC when the message buffer status MBS has changed and if bit MBI of that message buffer is set (see table 19.16). 0: No message buffer status change of message buffer with MBI = '1' 1: Message buffer status of at least one message buffer with MBI = '1' has changed
13	SUCS	0	R/W	Startup Completed Successfully This flag is set whenever a startup completed successfully and the CC entered NORMAL_ACTIVE state. 0: No startup completed successfully 1: Startup completed successfully

Bit	Bit Name	Initial Value	R/W	Description
12	SWE	0	R/W	Stop Watch Event This flag is set after a stop watch activation when the actual cycle counter and macrotick value are stored in the FlexRay Stop Watch Register. 0: No Stop Watch Event 1: Stop Watch Event occurred
11	TOBC	0	R/W	Transfer Output Buffer Completed This flag is set whenever a transfer from the Message RAM to the Output Buffer has completed and FROBCR.OBSYS has been reset by the Message Handler. 0: No transfer completed 1: Transfer between Message RAM and Output Buffer completed
10	TIBC	0	R/W	Transfer Input Buffer Completed This flag is set whenever a transfer from Input Buffer to the Message RAM has completed and FRIBCR.IBSYS has been reset by the Message Handler. 0: No transfer completed 1: Transfer between Input Buffer and Message RAM completed
9	TI1	0	R/W	Timer Interrupt 1 This flag is set whenever timer 1 matches the conditions configured in register FRT1C. The FT1IS flag in FXRTISR is also set to 1. 0: No Timer Interrupt 1 1: Timer Interrupt 1 occurred
8	TI0	0	R/W	Timer Interrupt 0 This flag is set whenever timer 0 matches the conditions configured in register FRT0C. The FT0IS flag in FXRTISR is also set to 1. 0: No Timer Interrupt 0 1: Timer Interrupt 0 occurred

Bit	Bit Name	Initial Value	R/W	Description
7	NMVC	0	R/W	Network Management Vector Changed This interrupt flag signals a change in the Network Management Vector visible to the Host. 0: No change in the network management vector 1: Network management vector changed
6	RFCL	0	R/W	Receive FIFO Critical Level This flag is set when the receive FIFO fill level FRFSR.RFFL[7:0] is equal or greater than the critical level as configured by FRFCL.CL[7:0]. 0: Receive FIFO below critical level 1: Receive FIFO critical level reached
5	RFNE	0	R/W	Receive FIFO Not Empty This flag is set by the CC when a received valid frame was stored into the empty receive FIFO. The actual state of the receive FIFO is monitored in register FRFSR. 0: Receive FIFO is empty 1: Receive FIFO is not empty
4	RXI	0	R/W	Receive Interrupt This flag is set by the CC whenever the set condition of a message buffers ND flag is fulfilled (see section 19.11.6, FlexRay New Data 1/2/3/4 (FRNDAT1/2/3/4)), and if bit MBI of that message buffer is set to '1' (see table 19.16). 0: No ND flag of a receive buffer with MBI = '1' has been set to '1' 1: At least one ND flag of a receive buffer with MBI = '1' has been set to '1'
3	TXI	0	R/W	Transmit Interrupt This flag is set by the CC at the end of frame transmission if bit MBI in the respective message buffer is set to '1' (see table 19.16). 0: No frame transmitted from a transmit buffer with MBI = '1' 1: At least one frame was transmitted from a transmit buffer with MBI = '1'

Bit	Bit Name	Initial Value	R/W	Description
2	CYCS	0	R/W	Cycle Start Interrupt This flag is set by the CC when a communication cycle starts. 0: No communication cycle started 1: Communication cycle started
1	CAS	0	R/W	Collision Avoidance Symbol This flag is set by the CC during STARTUP state when a CAS or a potential CAS was received. 0: No bit pattern matching the CAS symbol received 1: Bit pattern matching the CAS symbol received
0	WST	0	R/W	Wakeup Status This flag is set when the wakeup status vector FRCCSV.WSV[2:0] is changed by a protocol event. 0: Wakeup status unchanged 1: Wakeup status changed

19.7.3 FlexRay Error Interrupt Line Select (FREILS)

The FlexRay Error Interrupt Line Select register assigns an interrupt generated by a specific error interrupt flag from register FREIR to one of the two module interrupt lines:

1: Interrupt assigned to FlexRay1 interrupt request line
0: Interrupt assigned to FlexRay0 interrupt request line

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	TABBL	LTVBL	EDBL	—	—	—	—	—	TABAL	LTVAL	EDAL
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	MHFL	IOBAL	IIBAL	EFAL	RFOL	PERRL	CCLL	CCFL	SFOL	SFBML	CNAL	PEMCL
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
26	TABBL	0	R/W	Transmission Across Boundary Channel B Interrupt Line
25	LTVBL	0	R/W	Latest Transmit Violation Channel B Interrupt Line
24	EDBL	0	R/W	Error Detected on Channel B Interrupt Line
23 to 19	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
18	TABAL	0	R/W	Transmission Across Boundary Channel A Interrupt Line
17	LTVAL	0	R/W	Latest Transmit Violation Channel A Interrupt Line
16	EDAL	0	R/W	Error Detected on Channel A Interrupt Line
15 to 12	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
11	MHFL	0	R/W	Message Handler Constraints Flag Interrupt Line
10	IOBAL	0	R/W	Illegal Output Buffer Access Interrupt Line
9	IIBAL	0	R/W	Illegal Input Buffer Access Interrupt Line
8	EFAL	0	R/W	Empty FIFO Access Interrupt Line

Bit	Bit Name	Initial Value	R/W	Description
7	RFOL	0	R/W	Receive FIFO Overrun Interrupt Line
6	PERRL	0	R/W	Parity Error Interrupt Line
5	CCLL	0	R/W	CHI Command Locked Interrupt Line
4	CCFL	0	R/W	Clock Correction Failure Interrupt Line
3	SFOL	0	R/W	Sync Frame Overflow Interrupt Line
2	SFBML	0	R/W	Sync Frames Below Minimum Interrupt Line
1	CNAL	0	R/W	Command Not Accepted Interrupt Line
0	PEMCL	0	R/W	POC Error Mode Changed Interrupt Line

19.7.4 FlexRay Status Interrupt Line Select (FRSILS)

The FlexRay Status Interrupt Line Select register assigns an interrupt generated by a specific status interrupt flag from register FRSIR to one of the two module interrupt lines:

1: Interrupt assigned to FlexRay1 interrupt request line
0: Interrupt assigned to FlexRay0 interrupt request line

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	MTSBL	WUPBL	—	—	—	—	—	—	MTSAL	WUPAL
Initial value:	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1
R/W:	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SDSL	MBSIL	SUCSL	SWEL	TOBCL	TIBCL	TI1L	TI0L	NMVCL	RFCLL	RFNEL	RXIL	TXIL	CYCSL	CASL	WSTL
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 26	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
25	MTSBL	1	R/W	Media Access Test Symbol Channel B Interrupt Line
24	WUPBL	1	R/W	Wakeup Pattern Channel B Interrupt Line
23 to 18	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
17	MTSAL	1	R/W	Media Access Test Symbol Channel A Interrupt Line
16	WUPAL	1	R/W	Wakeup Pattern Channel A Interrupt Line
15	SDSL	1	R/W	Start of Dynamic Segment Interrupt Line
14	MBSIL	1	R/W	Message Buffer Status Interrupt Line
13	SUCSL	1	R/W	Startup Completed Successfully Interrupt Line
12	SWEL	1	R/W	Stop Watch Event Interrupt Line
11	TOBCL	1	R/W	Transfer Output Buffer Completed Interrupt Line
10	TIBCL	1	R/W	Transfer Input Buffer Completed Interrupt Line
9	TI1L	1	R/W	Timer Interrupt 1 Line
8	TI0L	1	R/W	Timer Interrupt 0 Line

Bit	Bit Name	Initial Value	R/W	Description
7	NMVCL	1	R/W	Network Management Vector Changed Interrupt Line
6	RFCLL	1	R/W	Receive FIFO Critical Level Interrupt Line
5	RFNEL	1	R/W	Receive FIFO Not Empty Interrupt Line
4	RXIL	1	R/W	Receive Interrupt Line
3	TXIL	1	R/W	Transmit Interrupt Line
2	CYCSL	1	R/W	Cycle Start Interrupt Line
1	CASL	1	R/W	Collision Avoidance Symbol Interrupt Line
0	WSTL	1	R/W	Wakeup Status Interrupt Line

19.7.5 FlexRay Error Interrupt Enable Set/Reset (FREIES, FREIER)

The settings in the FlexRay Error Interrupt Enable register determine which status changes in the FlexRay Error Interrupt Register will result in an interrupt.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	TABBE	LTVBE	EDBE	—	—	—	—	—	TABAE	LTVAE	EDAE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	MHFE	IOBAE	IIBAE	EFAE	RFOE	PERRE	CCLE	CCFE	SFOE	SFBME	CNAE	PEMCE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

To enable an error interrupt, write '1' to the corresponding bit in FREIES. To disable an interrupt, write '1' to the corresponding bit in FREIER. Writing '0' has no effect. Reading from both registers will result in the same value.

1: Interrupt enabled
0: Interrupt disabled

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
26	TABBE	0	R/W	Transmission Across Boundary Channel B Interrupt Enable
25	LTVBE	0	R/W	Latest Transmit Violation Channel B Interrupt Enable
24	EDBE	0	R/W	Error Detected on Channel B Interrupt Enable
23 to 19	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
18	TABAE	0	R/W	Transmission Across Boundary Channel A Interrupt Enable
17	LTVAE	0	R/W	Latest Transmit Violation Channel A Interrupt Enable
16	EDAE	0	R/W	Error Detected on Channel A Interrupt Enable

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
11	MHFE	0	R/W	Message Handler Constraints Flag Interrupt Enable
10	IOBAE	0	R/W	Illegal Output Buffer Access Interrupt Enable
9	IIBAE	0	R/W	Illegal Input Buffer Access Interrupt Enable
8	EFAE	0	R/W	Empty FIFO Access Interrupt Enable
7	RFOE	0	R/W	Receive FIFO Overrun Interrupt Enable
6	PERRE	0	R/W	Parity Error Interrupt Enable
5	CCLE	0	R/W	CHI Command Locked Interrupt Enable
4	CCFE	0	R/W	Clock Correction Failure Interrupt Enable
3	SFOE	0	R/W	Sync Frame Overflow Interrupt Enable
2	SFBME	0	R/W	Sync Frames Below Minimum Interrupt Enable
1	CNAE	0	R/W	Command Not Accepted Interrupt Enable
0	PEMCE	0	R/W	POC Error Mode Changed Interrupt Enable

19.7.6 FlexRay Status Interrupt Enable Set/Reset (FRSIES, FRSIER)

The settings in the FlexRay Status Interrupt Enable register determine which status changes in the FlexRay Status Interrupt Register will result in an interrupt.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	MTSBE	WUPBE	—	—	—	—	—	—	MTSAE	WUPAE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SDSE	MBSIE	SUCSE	SWEE	TOBCE	TIBCE	TI1E	TI0E	NMVCE	RFCLE	RFNEE	RXIE	TXIE	CYCSE	CASE	WSTE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

To enable a status interrupt, write '1' to the corresponding bit in FRSIES. To disable an interrupt, write '1' to the corresponding bit in FRSIER. Writing '0' has no effect. Reading from both registers will result in the same value.

1: Interrupt enabled
0: Interrupt disabled

Bit	Bit Name	Initial Value	R/W	Description
31 to 26	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
25	MTSBE	0	R/W	MTS Received on Channel B Interrupt Enable
24	WUPBE	0	R/W	Wakeup Pattern Channel B Interrupt Enable
23 to 18	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
17	MTSAE	0	R/W	MTS Received on Channel A Interrupt Enable
16	WUPAE	0	R/W	Wakeup Pattern Channel A Interrupt Enable
15	SDSE	0	R/W	Start of Dynamic Segment Interrupt Enable
14	MBSIE	0	R/W	Message Buffer Status Interrupt Enable
13	SUCSE	0	R/W	Startup Completed Successfully Interrupt Enable
12	SWEE	0	R/W	Stop Watch Event Interrupt Enable
11	TOBCE	0	R/W	Transfer Output Buffer Completed Interrupt Enable

Bit	Bit Name	Initial Value	R/W	Description
10	TIBCE	0	R/W	Transfer Input Buffer Completed Interrupt Enable
9	TI1E	0	R/W	Timer Interrupt 1 Enable
8	TI0E	0	R/W	Timer Interrupt 0 Enable
7	NMVCE	0	R/W	Network Management Vector Changed Interrupt Enable
6	RFCLE	0	R/W	Receive FIFO Critical Level Interrupt Enable
5	RFNEE	0	R/W	Receive FIFO Not Empty Interrupt Enable
4	RXIE	0	R/W	Receive Interrupt Enable
3	TXIE	0	R/W	Transmit Interrupt Enable
2	CYCSE	0	R/W	Cycle Start Interrupt Enable
1	CASE	0	R/W	Collision Avoidance Symbol Interrupt Enable
0	WSTE	0	R/W	Wakeup Status Interrupt Enable

19.7.7 FlexRay Interrupt Line Enable (FRILE)

Each of the two interrupt lines to the Host (FlexRay0 and FlexRay1 interrupt requests) can be enabled/disabled separately by programming bit EINT0 and EINT1.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	EINT1	EINT0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
1	EINT1	0	R/W	Enable Interrupt Line 1 0: FlexRay1 interrupt request disabled 1: FlexRay1 interrupt request enabled
0	EINT0	0	R/W	Enable Interrupt Line 0 0: FlexRay0 interrupt request disabled 1: FlexRay0 interrupt request enabled

19.7.8 FlexRay Timer 0 Configuration (FRT0C)

Absolute timer. Specifies in terms of cycle count and macrotick the point in time when the timer 0 interrupt occurs. When a timer 0 interrupt occurs, the TI0 bit in FRSIR and FT0IS bit in FXRTISR are set to '1'.The timer 0 interrupt request continues to be effective while the FlexRay timer 0 interrupt enable bit (FT0IEN) is set to '1'.

Timer 0 can be activated as long as the POC is either in NORMAL_ACTIVE state or in NORMAL_PASSIVE state. Timer 0 is deactivated when leaving NORMAL_ACTIVE state or NORMAL_PASSIVE state except for transitions between the two states.

Before reconfiguration of the timer, the timer has to be halted first by writing bit T0RC to '0'.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	T0MO[13:0]													
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	T0CC[6:0]							—	—	—	—	—	—	T0MS	T0RC
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31, 30	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
29 to 16	T0MO[13:0]	All 0	R/W	Timer 0 Macrotick Offset Configures the macrotick offset from the beginning of the cycle where the interrupt is to occur. The Timer 0 Interrupt occurs at this offset for each cycle of the cycle set.
15	—	0	R	Reserved. This bit is always read as 0. The write value should always be 0.
14 to 8	T0CC[6:0]	All 0	R/W	Timer 0 Cycle Code The 7-bit timer 0 cycle code determines the cycle set used for generation of the timer 0 interrupt. For details about the configuration of the cycle code see section 19.20.2, Cycle Counter Filtering.

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
1	T0MS	0	R/W	Timer 0 Mode Select 0: Single-shot mode 1: Continuous mode
0	T0RC	0	R/W	Timer 0 Run Control 0: Timer 0 halted 1: Timer 0 running

Note: The configuration of timer 0 is compared against the macrotick counter value; there is no separate counter for timer 0.

19.7.9 FlexRay Timer 1 Configuration (FRT1C)

Relative timer. When the specified number of macroticks has expired, the TI1 bit in FRSIR and the FT1IS bit in FXRTISR are set to '1'. The timer 1 interrupt request continues to be effective while the FlexRay timer 1 interrupt enable bit (FT1IEN) is set to '1'.

Timer 1 can be activated as long as the POC is either in NORMAL_ACTIVE state or in NORMAL_PASSIVE state. Timer 1 is deactivated when leaving NORMAL_ACTIVE state or NORMAL_PASSIVE state except for transitions between the two states.

Before reconfiguration of the timer, the timer has to be halted first by writing bit T1RC to '0'.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	T1MC[13:0]													
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	T1MS	T1RC
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31, 30	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
29 to 16	T1MC[13:0]	H'0002	R/W	Timer 1 Macrotick Count When the configured macrotick count is reached the timer 1 interrupt is generated. Valid values are: 2 to 16383 MT in continuous mode 1 to 16383 MT in single-shot mode
15 to 2	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
1	T1MS	0	R/W	Timer 1 Mode Select 0: Single-shot mode 1: Continuous mode

Bit	Bit Name	Initial Value	R/W	Description
0	T1RC	0	R/W	Timer 1 Run Control 0: Timer 1 halted 1: Timer 1 running

19.7.10 FlexRay Stop Watch Register 1 (FRSTPW1)

The stop watch is activated by a FlexRay0 or FlexRay1interrupt request or by the Host by writing bit SSWT to '1'.

With the macrotick counter increment following next to the stop watch activation the actual cycle counter and macrotick values are captured in register FRSTPW1 while the slot counter values for channel A and B are captured in register FRSTPW2.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	SMTV[13:0]													
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	SCCV[5:0]						—	EINT1	EINT0	—	SSWT	—	SWMS	ESWT
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31, 30	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
29 to 16	SMTV[13:0]	All 0	R	Stop Watch Captured Macrotick Value State of the macrotick counter when the stop watch event occurred. Valid values are 0 to 15999.
15, 14	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
13 to 8	SCCV[5:0]	All 0	R	Stop Watch Captured Cycle Counter Value State of the cycle counter when the stop watch event occurred. Valid values are 0 to 63.
7	—	0	R	Reserved. This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
6	EINT1	0	R/W	FlexRay1 Interrupt Trigger Enable Enables stop watch trigger by FlexRay1 interrupt event if ESWT = '1'. 0: Stop watch trigger by FlexRay1 interrupt disabled 1: FlexRay1 Interrupt event triggers stop watch
5	EINT0	0	R/W	FlexRay0 Interrupt Trigger Enable Enables stop watch trigger by FlexRay0 Interrupt event if ESWT = '1'. 0: Stop watch trigger by FlexRay0 interrupt disabled 1: FlexRay0 interrupt event triggers stop watch
4	—	0	R	Reserved. This bit is always read as 0. The write value should always be 0.
3	SSWT	0	R/W	Software Stop Watch Trigger When the Host writes this bit to '1' the stop watch is activated. After the actual cycle counter and macrotick value are stored in the FlexRay Stop Watch Register this bit is reset to '0'. The bit is only writeable while ESWT = '0'. 0: Software trigger reset 1: Stop watch activated by software trigger
2	—	0	R	Reserved. This bit is always read as 0. The write value should always be 0.
1	SWMS	0	R/W	Stop Watch Mode Select 0: Single-shot mode 1: Continuous mode
0	ESWT	0	R/W	Enable Stop Watch Trigger If enabled an interrupt 0,1 event (rising edge on FlexRay0 or FlexRay1 interrupt request pin) activates the stop watch. In single-shot mode this bit is reset to '0' after the actual cycle counter and macrotick value are stored in the FlexRay Stop Watch Register. 0: Stop watch trigger disabled 1: Stop watch trigger enabled

Note: 1. Bits ESWT and SSWT cannot be set to '1' simultaneously. In this case the write access to the register is ignored, and both bits keep their previous values. Either the external stop watch trigger or the software stop watch trigger may be used.

2. Be sure to write 0 to bits 4 and 2.

19.7.11 FlexRay Stop Watch Register 2 (FRSTPW2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	SSCVB[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SSCVA[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
26 to 16	SSCVB[10:0]	All 0	R	Stop Watch Captured Slot Counter Value Channel B State of the slot counter for channel B when the stop watch event occurred. Valid values are 0 to 2047.
15 to 11	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
10 to 0	SSCVA[10:0]	All 0	R	Stop Watch Captured Slot Counter Value Channel A State of the slot counter for channel A when the stop watch event occurred. Valid values are 0 to 2047.

19.8 CC Control Registers

This section describes the registers provided by the CC to allow the Host to control the operation of the CC. The FlexRay protocol specification requires the Host to write application configuration data in CONFIG state only. Please consider that the configuration registers are not locked for writing in DEFAULT_CONFIG state.

The configuration data is reset when DEFAULT_CONFIG state is entered from hardware or software reset. To change POC state from DEFAULT_CONFIG to CONFIG state the Host has to apply CHI command CONFIG. If the Host wants the CC to leave CONFIG state, the Host has to proceed as described in section 19.6.5, FlexRay Lock Register (FRLCK).

19.8.1 FlexRay SUC Configuration Register 1 (FRSUCC1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	CCHB	CCHA	MTSB	MTSA	HCSE	TSM	WUCS	PTA[4:0]				
Initial value:	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSA[4:0]					—	TXSY	TXST	PBSY	—	—	—	CMD[3:0]			
Initial value:	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
27	CCHB	1	R/W	Connected to Channel B (pChannels) Configures whether the node is connected to channel B. 0: Not connected to channel B 1: Node connected to channel B (the default value after hardware or software reset)
26	CCHA	1	R/W	Connected to Channel A (pChannels) Configures whether the node is connected to channel A. 0: Not connected to channel A 1: Node connected to channel A (the default value after hardware or software reset)

Bit	Bit Name	Initial Value	R/W	Description
25	MTSB	0	R/W	Select Channel B for MTS Transmission * The bit selects channel B for MTS symbol transmission. The flag is reset by default and may be modified only in DEFAULT_CONFIG or CONFIG state. 0: Channel B disabled for MTS transmission 1: Channel B selected for MTS transmission
24	MTSA	0	R/W	Select Channel A for MTS Transmission * The bit selects channel A for MTS symbol transmission. The flag is reset by default and may be modified only in DEFAULT_CONFIG or CONFIG state. 0: Channel A disabled for MTS transmission 1: Channel A selected for MTS transmission
23	HCSE	0	R/W	Halt due to Clock Sync Error (pAllowHaltDueToClock) Controls the transition to HALT state due to a clock synchronization error. The bit can be modified in DEFAULT_CONFIG or CONFIG state only. 0: CC will enter/remain in NORMAL_PASSIVE 1: CC will enter HALT state

Bit	Bit Name	Initial Value	R/W	Description
22	TSM	1	R/W	Transmission Slot Mode (pSingleSlotEnabled) Selects the initial transmission slot mode. In SINGLE slot mode the CC may only transmit in the preconfigured key slot. The key slot ID is configured in the header section of message buffer 0 respectively message buffers 0 and 1 depending on bit FRMRC.SPLM. In case TSM = '1', message buffer 0 respectively message buffers 0,1 can be (re)configured in DEFAULT_CONFIG or CONFIG state only. In ALL slot mode the CC may transmit in all slots. TSM is a configuration bit which can only be set/reset by the Host. The bit can be written in DEFAULT_CONFIG or CONFIG state only. The CC changes to ALL slot mode when the Host successfully applied the ALL_SLOTS command by writing CMD[3:0] = "0101" in POC state NORMAL_ACTIVE or NORMAL_PASSIVE. The actual slot mode is monitored by FRCCSV.SLM[1:0]. The bit can be modified in DEFAULT_CONFIG or CONFIG state only. 0: ALL Slot Mode 1: SINGLE Slot Mode (the default value after hardware or software reset)
21	WUCS	0	R/W	Wakeup Channel Select (pWakeupChannel) With this bit the Host selects the channel on which the CC sends the Wakeup pattern. The CC ignores any attempt to change the status of this bit when not in DEFAULT_CONFIG or CONFIG state. 0: Send wakeup pattern on channel A 1: Send wakeup pattern on channel B
20 to 16	PTA[4:0]	All 0	R/W	Passive to Active (pAllowPassiveToActive) Defines the number of consecutive even/odd cycle pairs that must have valid clock correction terms before the CC is allowed to transit from NORMAL_PASSIVE to NORMAL_ACTIVE state. It can be modified in DEFAULT_CONFIG or CONFIG state only. Valid values are 0 to 31 even/odd cycle pairs. If set to "00000" the CC is not allowed to transit from NORMAL_PASSIVE to NORMAL_ACTIVE state.

Bit	Bit Name	Initial Value	R/W	Description
15 to 11	CSA[4:0]	H'02	R/W	Cold Start Attempts (gColdStartAttempts) Configures the maximum number of attempts that a cold starting node is permitted to try to start up the network without receiving any valid response from another node. It can be modified in DEFAULT_CONFIG or CONFIG state only. Must be identical in all nodes of a cluster. Valid values are 2 to 31.
10	—	0	R	Reserved. This bit is always read as 0. The write value should always be 0.
9	TXSY	0	R/W	Transmit Sync Frame in Key Slot (pKeySlotUsedForSync) Defines whether the key slot is used to transmit sync frames. The protocol requires that both TXST and TXSY bits are set for coldstart nodes. The bit can be modified in DEFAULT_CONFIG or CONFIG state only. 0: No sync frame transmission in key slot, node is neither sync nor coldstart node 1: Key slot used to transmit sync frame, node is sync node
8	TXST	0	R/W	Transmit Startup Frame in Key Slot (pKeySlotUsedForStartup) Defines whether the key slot is used to transmit startup frames. The protocol requires that both TXST and TXSY bits are set for coldstart nodes. The bit can be modified in DEFAULT_CONFIG or CONFIG state only. 0: No startup frame transmission in key slot, node is non-coldstarter 1: Key slot used to transmit startup frame, node is leading or following coldstarter

Bit	Bit Name	Initial Value	R/W	Description
7	PBSY	1	R	POC Busy Signals that the POC is busy and cannot accept a command from the Host. CMD[3:0] is locked against write accesses. Set to '1' after hardware or software reset during initialization of internal RAM blocks. 0: POC not busy, CMD[3:0] writeable 1: POC is busy, CMD[3:0] locked
6 to 4	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
3 to 0	CMD[3:0]	All 0	R/W	CHI Command Vector The Host may write any CHI command at any time, but certain commands are enabled only in certain POC states. If a command is not enabled, it will not be executed, the CHI command vector CMD[3:0] will be reset to "0000" = command_not_accepted, and flag FREIR.CNA will be set to '1'. In case the previous CHI command has not yet completed, FREIR.CCL is set to '1' together with FREIR.CNA; the CHI command needs to be repeated. Except for HALT state, a POC state change command applied while the CC is already in the requested POC state neither causes a state change nor will FREIR.CNA flag be set to 1. 0000: command_not_accepted 0001: CONFIG 0010: READY 0011: WAKEUP 0100: RUN 0101: ALL_SLOTS 0110: HALT 0111: FREEZE 1000: SEND_MTS 1001: ALLOW_COLDSTART 1010: RESET_STATUS_INDICATORS 1011: Reserved 1100: CLEAR_RAMS 1101 to 1111: Reserved

Bit	Bit Name	Initial Value	R/W	Description
3 to 0	CMD[3:0]	All 0	R/W	Reading CMD[3:0] shows whether the last CHI command was accepted. The actual POC state is monitored by FRCCSV.POCS[5:0]. In general the Host must check FRSUCC1.PBSY before writing a new CHI command. Note: When the reserved CHI command value is specified in CMD[3:0], subsequent operation cannot be guaranteed. command_not_accepted: CMD[3:0] is reset to "0000" due to one of the following conditions: • Illegal command applied by the Host • Host applied command to leave CONFIG state without preceding configuration lock key • Host applied new command while execution of the previous Host command has not completed • Host writes command_not_accepted When CMD[3:0] is reset to "0000", FREIR.CNA is set, and, if enabled, an interrupt is generated. Commands which are not accepted are not executed. CONFIG Go to POC state CONFIG when called in POC state DEFAULT_CONFIG or READY. When called in HALT state the CC transits to POC state DEFAULT_CONFIG. When called in any other state, CMD[3:0] will be reset to "0000" = command_not_accepted. READY Go to POC state READY when called in POC state CONFIG, NORMAL_ACTIVE, NORMAL_PASSIVE, STARTUP, or WAKEUP. When called in any other state, CMD[3:0] will be reset to "0000" = command_not_accepted. WAKEUP Go to POC state WAKEUP when called in POC state READY. When called in any other state, CMD[3:0] will be reset to "0000" = command_not_accepted. RUN Go to POC state STARTUP when called in POC state READY. When called in any other state, CMD[3:0] will be reset to "0000" = command_not_accepted.

Bit	Bit Name	Initial Value	R/W	Description
3 to 0	CMD[3:0]	All 0	R/W	ALL_SLOTS Leave SINGLE slot mode after successful startup/integration at the next end of cycle when called in POC state NORMAL_ACTIVE or NORMAL_PASSIVE. When called in any other state, CMD[3:0] will be reset to "0000" = command_not_accepted. HALT Set halt request FRCCSV.HRQ and go to POC state HALT at the next end of cycle when called in POC state NORMAL_ACTIVE or NORMAL_PASSIVE. When called in any other state, CMD[3:0] will be reset to "0000" = command_not_accepted. FREEZE Set the freeze status indicator FRCCSV.FSI and go to POC state HALT immediately. Can be called from any state. SEND_MTS Send single MTS symbol during the next following symbol window on the channel configured by MTSA, MTSB, when called in POC state NORMAL_ACTIVE after CC entered ALL slot mode (FRCCSV.SLM[1:0] = "11"). When called in any other state, or when called while a previously requested MTS has not yet been transmitted, CMD[3:0] will be reset to "0000" = command_not_accepted. ALLOW_COLDSTART The command resets FRCCSV.CSI to enable the node to become leading coldstarter. When called in state DEFAULT_CONFIG, CONFIG, or HALT, CMD[3:0] will be reset to "0000" = command_not_accepted. To become leading coldstarter it is also required that both TXST and TXSY are set. RESET_STATUS_INDICATORS Resets status flags FRCCSV.CSNI, FRCCSV.CSAI, and FRCCSV.WSV[2:0] to their default values. May be called in POC states READY and STARTUP. When called in any other state, CMD[3:0] will be reset to "0000" = command_not_accepted. CLEAR_RAMS Sets FRMHDS.CRAM when called in DEFAULT_CONFIG or CONFIG state. When called in any other state, CMD[3:0] will be reset to "0000" = command_not_accepted.

Bit	Bit Name	Initial Value	R/W	Description
3 to 0	CMD[3:0]	All 0	R/W	FRMHDS.CRAM is also set when the CC leaves hardware or software reset. By setting FRMHDS.CRAM all internal RAM blocks are initialized to zero. During the initialization of the RAMs, PBSY will show POC busy. Access to the configuration and status registers is possible during execution of CHI command CLEAR_RAMS. The initialization of the FlexRay internal RAM blocks requires 2048 peripheral bus clock cycles. There should be no Host access to IBF or OBF during initialization of the internal RAM blocks after hardware or software reset or after assertion of CHI command CLEAR_RAMS. Before asserting CHI command CLEAR_RAMS the Host should make sure that no transfer between Message RAM and IBF/OBF or the Transient Buffer RAMs is ongoing. This command also resets the FlexRay Message Buffer Status registers FRMHDS, FRLDTS, FRFSR, FRMHDF, FRTXRQ1/2/3/4, FRNDAT1/2/3/4, and FRMBSC1/2/3/4. Note: All accepted commands with exception of CLEAR_RAMS and SEND_MTS will cause a change of the POC state in the sampling clock domain after at most 8 cycles of the slower of the two clocks peripheral bus clock and sampling clock, counted from the falling edge of the FlexRay module select, assumed that POC was not busy when the command was applied and that no POC state change was forced by bus activity in that time frame. Reading register FRCCSV will show data that is additionally delayed by synchronization from sampling clock to peripheral bus clock domain and by the Host-specific CPU interface. The maximum additional delay is 12 cycles of the slower of the two clocks peripheral bus clock and sampling clock.

Note: * MTSA and MTSB may also be changed outside DEFAULT_CONFIG or CONFIG state when the write to FRSUCC1 register is directly preceded by the unlock sequence for the Configuration Lock Key as described in section 19.6.5, FlexRay Lock Register (FRLCK). This may be combined with CHI command SEND_MTS. If both bits MTSA and MTSB are set to '1' an MTS symbol will be transmitted on both channels when requested by writing CMD[3:0] = "1000".

Table 19.3 references the CHI commands from the FlexRay Protocol Specification v2.1 (section 2.2.1.1, Table 2-2) to the FlexRay CHI command vector CMD[3:0].

Table 19.3 Reference to CHI Host Command Summary from FlexRay Protocol Specification

CHI command	Where processed (POC States)	CHI Command Vector CMD[3:0]
ALL_SLOTS	POC: normal active, POC: normal passive	ALL_SLOTS
ALLOW_COLDSTART	All except POC: default config, POC: config, POC: halt	ALLOW_COLDSTART
CONFIG	POC: default config, POC: ready	CONFIG
CONFIG_COMPLETE	POC: config	Unlock sequence & READY
DEFAULT_CONFIG	POC: halt	CONFIG
FREEZE	All	FREEZE
HALT	POC: normal active, POC: normal passive	HALT
READY	READY All except POC: default config, POC: config, POC: ready, POC: halt	READY
RUN	POC: ready	RUN
WAKEUP	POC: ready	WAKEUP

19.8.2 FlexRay SUC Configuration Register 2 (FRSUCC2)

The CC accepts modifications of the register in DEFAULT_CONFIG or CONFIG state only.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	LTN[3:0]*				—	—	—	LT[20:16]*				
Initial value:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LT[15:0]*															
Initial value:	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
27 to 24	LTN[3:0]	0001	R/W	Listen Timeout Noise (gListenNoise – 1) Configures the upper limit for startup and wakeup listen timeout in the presence of noise expressed as a multiple of pdListenTimeout. LTN[3:0] must be configured identical in all nodes of a cluster. The range for gListenNoise is 2 to 16.
23 to 21	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
20 to 0	LT[20:0]	H'000504	R/W	Listen Timeout (pdListenTimeout) Configures wakeup/startup listen timeout in μT. The range for pdListenTimeout is 1284 to 1283846 μT.

Note: The wakeup/startup noise timeout is calculated as follows:
pdListenTimeout × gListenNoise = LT20 to LT0 bits × (LTN3 to LTN0 bits +1)

19.8.3 FlexRay SUC Configuration Register 3 (FRSUCC3)

The CC accepts modifications of the register in DEFAULT_CONFIG or CONFIG state only.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	WCF[3:0]*				WCP[3:0]*			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
7 to 4	WCF[3:0]	0001	R/W	Maximum Without Clock Correction Fatal (gMaxWithoutClockCorrectionFatal) Defines the number of consecutive even/odd cycle pairs with missing clock correction terms that will cause a transition from NORMAL_ACTIVE or NORMAL_PASSIVE to HALT state. Must be identical in all nodes of a cluster. Valid values are 1 to 15 cycle pairs. Note: The transition to HALT state is prevented if the HCSE bit in FRSUCC1 is not set to 1.
3 to 0	WCP[3:0]	0001	R/W	Maximum Without Clock Correction Passive (gMaxWithoutClockCorrectionPassive) Defines the number of consecutive even/odd cycle pairs with missing clock correction terms that will cause a transition from NORMAL_ACTIVE to NORMAL_PASSIVE state. Must be identical in all nodes of a cluster. Valid values are 1 to 15 cycle pairs.

19.8.4 FlexRay NEM Configuration Register (FRNEMC)

The CC accepts modifications of the register in DEFAULT_CONFIG or CONFIG state only.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	NML[3:0]*			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
3 to 0	NML[3:0]	All 0	R/W	Network Management Vector Length (gNetworkManagementVectorLength) These bits configure the length of the NM vector. The configured length must be identical in all nodes of a cluster. Valid values are 0 to 12 bytes.

19.8.5 FlexRay PRT Configuration Register 1 (FRPRTC1)

The CC accepts modifications of the register in DEFAULT_CONFIG or CONFIG state only.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RWP[5:0]*						—	RXW[8:0]*								
Initial value:	0	0	0	0	1	0	0	0	0	1	0	0	1	1	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	BRP0*	SPP1*	SPP0*	—	CASM6	CASM[5:0]*						TSST[3:0]*			
Initial value:	0	0	0	0	0	1	1	0	0	0	1	1	0	0	1	1
R/W:	R	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 26	RWP[5:0]	H'02	R/W	Repetitions of Tx Wakeup Pattern (pWakeupPattern) Configures the number of repetitions (sequences) of the Tx wakeup symbol. Valid values are 2 to 63.
25	—	0	R	Reserved. This bit is always read as 0. The write value should always be 0.
24 to 16	RXW[8:0]	H'04C	R/W	Wakeup Symbol Receive Window Length (gdWakeupSymbolRxWindow) Configures the number of bit times used by the node to test the duration of the received wakeup pattern. Must be identical in all nodes of a cluster. Valid values are 76 to 301 bit times.
15	—	0	R	Reserved. This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
14	BRP0	0	R/W	Baud Rate Prescaler (gdSampleClockPeriod, pSamplesPerMicrotick) The Baud Rate Prescaler configures the baud rate on the FlexRay bus. The baud rates listed below are valid with FlexRay frequency = 80 MHz. One bit time always consists of 8 samples independent of the configured baud rate. 0: 10 Mbps operation (default) gdSampleClockPeriod = 12.5 ns pSamplePerMicrotick = 2 (1μT = 25 ns) 1: 5 Mbps operation gdSampleClockPeriod = 25 ns pSamplePerMicrotick = 1 (1μT = 25 ns) Note: Be sure to write 0 to bit 15.
13	SPP1	0	R/W	Strobe Point Position Defines the sample count value for strobing. The strobed bit value is set to the voted value when the sample count is incremented to the value configured by SPP[1:0]. 00: Sample 5 (default) 01: Sample 4 10: Sample 6 11: Sample 5 Note: The current revision 2.1 of the FlexRay protocol requires that SPP[1:0] = "00". The alternate strobe point positions could be used to compensate for asymmetries in the physical layer.
12	SPP0	0	R/W	
11	—	0	R	Reserved. This bit is always read as 0. The write value should always be 0.
10	CASM6	1	R	Collision Avoidance Symbol Max (gdCASRxLowMax) Configures the upper limit of the acceptance window for a collision avoidance symbol (CAS). CASM6 is fixed to '1'. Valid values are 67 to 99 bit times.
9 to 4	CASM[5:0]	100011	R/W	

Bit	Bit Name	Initial Value	R/W	Description
3 to 0	TSST[3:0]	0011	R/W	Transmission Start Sequence Transmitter (gdTSSTransmitter) Configures the duration of the Transmission Start Sequence (TSS) in terms of bit times. Must be identical in all nodes of a cluster. 1 bit time = 4 μT = 100 ns @10 Mbps Valid values are 3 to 15 bit times.

19.8.6 FlexRay PRT Configuration Register 2 (FRPRTC2)

The CC accepts modifications of the register in DEFAULT_CONFIG or CONFIG state only.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	TXL[5:0]*						TXI[7:0]*							
Initial value:	0	0	0	0	1	1	1	1	0	0	1	0	1	1	0	1
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	RXL[5:0]*						—	—	RXI[5:0]*					
Initial value:	0	0	0	0	1	0	1	0	0	0	0	0	1	1	1	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31, 30	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
29 to 24	TXL[5:0]	H'0F	R/W	Wakeup Symbol Transmit Low (gdWakeupSymbolTxLow) Configures the number of bit times used by the node to transmit the low phase of the wakeup symbol. Must be identical in all nodes of a cluster. Valid values are 15 to 60 bit times.
23 to 16	TXI[7:0]	H'2D	R/W	Wakeup Symbol Transmit Idle (gdWakeupSymbolTxIdle) Configures the number of bit times used by the node to transmit the idle phase of the wakeup symbol. Must be identical in all nodes of a cluster. Valid values are 45 to 180 bit times.
15, 14	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
13 to 8	RXL[5:0]	H'0A	R/W	Wakeup Symbol Receive Low (gdWakeupSymbolRxLow) Configures the number of bit times used by the node to test the duration of the low phase of the received wakeup symbol. Must be identical in all nodes of a cluster. Valid values are 10 to 55 bit times.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
5 to 0	RXI[5:0]	H'0E	R/W	Wakeup Symbol Receive Idle (gdWakeupSymbolRxIdle) Configures the number of bit times used by the node to test the duration of the idle phase of the received wakeup symbol. Must be identical in all nodes of a cluster. Valid values are 14 to 59 bit times.

19.8.7 FlexRay MHD Configuration Register (FRMHDC)

The CC accepts modifications of the register in DEFAULT_CONFIG or CONFIG state only.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	SLT[12:0]*												
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	SFDL[6:0]*						
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
28 to 16	SLT[12:0]	All 0	R/W	Start of Latest Transmit (pLatestTx) Configures the maximum minislot value allowed before inhibiting frame transmission in the dynamic segment of the cycle. There is no transmission in dynamic segment if SLT[12:0] is set to zero. Valid values are 0 to 7981 minislots.
15 to 7	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
6 to 0	SFDL[6:0]	All 0	R/W	Static Frame Data Length (gPayloadLengthStatic) Configures the cluster-wide payload length for all frames sent in the static segment in double bytes. The payload length must be identical in all nodes of a cluster. Valid values are 0 to 127.

19.8.8 FlexRay GTU Configuration Register 1 (FRGTUC1)

The CC accepts modifications of the register in DEFAULT_CONFIG or CONFIG state only.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	UT[19:16]*			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UT[15:0]*															
Initial value:	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 20	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
19 to 0	UT[19:0]	H'00280	R/W	Microtick per Cycle (pMicroPerCycle) Configures the duration of the communication cycle in microticks. Valid values are 640 to 640000 μT.

19.8.9 FlexRay GTU Configuration Register 2 (FRGTUC2)

The CC accepts modifications of the register in DEFAULT_CONFIG or CONFIG state only.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	SNM[3:0]*			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	MPC[13:0]*													
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 20	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
19 to 16	SNM[3:0]	H'2	R/W	Sync Node Max (gSyncNodeMax) Maximum number of frames within a cluster with sync frame indicator bit SYN set to '1'. Must be identical in all nodes of a cluster. Valid values are 2 to 15.
15, 14	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
13 to 0	MPC[13:0]	H'000A	R/W	Macrotick Per Cycle (gMacroPerCycle) Configures the duration of one communication cycle in macroticks. The cycle length must be identical in all nodes of a cluster. Valid values are 10 to 16000 MT

19.8.10 FlexRay GTU Configuration Register 3 (FRGTUC3)

The CC accepts modifications of the register in DEFAULT_CONFIG or CONFIG state only.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	MIOB[6:0]*							—	MIOA[6:0]*						
Initial value:	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UIOB[7:0]*								UIOA[7:0]*							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	—	0	R	Reserved. This bit is always read as 0. The write value should always be 0.
30 to 24	MIOB[6:0]	H'02	R/W	Macrotick Initial Offset Channel B (pMacroInitialOffset[B]) Configures the number of macroticks between the static slot boundary and the subsequent macrotick boundary of the secondary time reference point based on the nominal macrotick duration. Must be identical in all nodes of a cluster. Valid values are 2 to 72 MT.
23	—	0	R	Reserved. This bit is always read as 0. The write value should always be 0.
22 to 16	MIOA[6:0]	H'02	R/W	Macrotick Initial Offset Channel A (pMacroInitialOffset[A]) Configures the number of macroticks between the static slot boundary and the subsequent macrotick boundary of the secondary time reference point based on the nominal macrotick duration. Must be identical in all nodes of a cluster. Valid values are 2 to 72 MT.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	UIOB[7:0]	All 0	R/W	Microtick Initial Offset Channel B (pMicroInitialOffset[B]) Configures the number of microticks between the actual time reference point on channel B and the subsequent macrotick boundary of the secondary time reference point. The parameter depends on the DCB[7:0] value in FRGTUC5 and therefore has to be set for each channel independently. Valid values are 0 to 240 μT.
7 to 0	UIOA[7:0]	All 0	R/W	Microtick Initial Offset Channel A (pMicroInitialOffset[A]) Configures the number of microticks between the actual time reference point on channel A and the subsequent macrotick boundary of the secondary time reference point. The parameter depends on the DCA[7:0] value in FRGTUC5 and therefore has to be set for each channel independently. Valid values are 0 to 240 μT.

19.8.11 FlexRay GTU Configuration Register 4 (FRGTUC4)

The CC accepts modifications of the register in DEFAULT_CONFIG or CONFIG state only. For details about configuration of NIT[13:0] and OCS[13:0] see section 19.14.5, Configuration of NIT Start and Offset Correction Start.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	OCS[13:0]*													
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	NIT[13:0]*													
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31, 30	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
29 to 16	OCS[13:0]	H'0008	R/W	Offset Correction Start (gOffsetCorrectionStart – 1) Determines the start of the offset correction within the NIT phase, calculated from start of cycle. Must be identical in all nodes of a cluster. Valid values are 8 to 15998 MT.
15, 14	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
13 to 0	NIT[13:0]	H'0007	R/W	Network Idle Time Start (gMacroPerCycle –gdNIT – 1) Configures the starting point of the Network Idle Time NIT at the end of the communication cycle expressed in terms of macroticks from the beginning of the cycle. Must be identical in all nodes of a cluster. Valid values are 7 to 15997 MT. The start of NIT is recognized if Macrotick = NIT[13:0] value and the increment pulse of Macrotick is set.

19.8.12 FlexRay GTU Configuration Register 5 (FRGTUC5)

The CC accepts modifications of the register in DEFAULT_CONFIG or CONFIG state only.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DEC[7:0]*								—	—	—	CDD[4:0]*				
Initial value:	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DCB[7:0]*								DCA[7:0]*							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	DEC[7:0]	H'0E	R/W	Decoding Correction (pDecodingCorrection) Configures the decoding correction value used to determine the primary time reference point. Valid values are 14 to 143 μT.
23 to 21	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
20 to 16	CDD[4:0]	All 0	R/W	Cluster Drift Damping (pClusterDriftDamping) Configures the cluster drift damping value used in clock synchronization to minimize accumulation of rounding errors. Valid values are 0 to 20 μT.
15 to 8	DCB[7:0]	All 0	R/W	Delay Compensation Channel B (pDelayCompensation[B]) Used to compensate for reception delays on the indicated channel. This covers assumed propagation delay up to cPropagationDelayMax for microticks in the range of 0.0125 to 0.05 μs. In practice, the minimum of the propagation delays of all sync nodes should be applied. Valid values are 0 to 200 μT.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	DCA[7:0]	All 0	R/W	Delay Compensation Channel A (pDelayCompensation[A]) Used to compensate for reception delays on the indicated channel. This covers assumed propagation delay up to cPropagationDelayMax for microticks in the range of 0.0125 to 0.05 µs. In practice, the minimum of the propagation delays of all sync nodes should be applied. Valid values are 0 to 200 µT.

19.8.13 FlexRay GTU Configuration Register 6 (FRGTUC6)

The CC accepts modifications of the register in DEFAULT_CONFIG or CONFIG state only.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	MOD[10:0]*										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	ASR[10:0]*										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
26 to 16	MOD[10:0]	H'002	R/W	Maximum Oscillator Drift (pdMaxDrift) Maximum drift offset between two nodes that operate with unsynchronized clocks over one communication cycle in µT. Valid values are 2 to 1923 µT.
15 to 11	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
10 to 0	ASR[10:0]	All 0	R/W	Accepted Startup Range (pdAcceptedStartupRange) Number of microticks constituting the expanded range of measured deviation for startup frames during integration. Valid values are 0 to 1875 µT.

19.8.14 FlexRay GTU Configuration Register 7 (FRGTUC7)

The CC accepts modifications of the register in DEFAULT_CONFIG or CONFIG state only.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	NSS[9:0]*									
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	SSL[9:0]*									
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 26	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
25 to 16	NSS[9:0]	H'002	R/W	Number of Static Slots (gNumberOfStaticSlots) Configures the number of static slots in a cycle. At least 2 coldstart nodes must be configured to startup a FlexRay network. The number of static slots must be identical in all nodes of a cluster. Valid values are 2 to 1023 MT.
15 to 10	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
9 to 0	SSL[9:0]	H'004	R/W	Static Slot Length (gdStaticSlot) Configures the duration of a static slot in macroticks. The static slot length must be identical in all nodes of a cluster. Valid values are 4 to 659 MT.

19.8.15 FlexRay GTU Configuration Register 8 (FRGTUC8)

The CC accepts modifications of the register in DEFAULT_CONFIG or CONFIG state only.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	NMS[12:0]*												
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	MSL[5:0]*					
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
28 to 16	NMS[12:0]	All 0	R/W	Number of Minislots (gNumberOfMinislots) Configures the number of minislots within the dynamic segment of a cycle. The number of minislots must be identical in all nodes of a cluster. Valid values are 0 to 7986 MT.
15 to 6	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
5 to 0	MSL[5:0]	H'02	R/W	Minislot Length (gdMinislot) Configures the duration of a minislot in macroticks. The minislot length must be identical in all nodes of a cluster. Valid values are 2 to 63 MT.

19.8.16 FlexRay GTU Configuration Register 9 (FRGTUC9)

The CC accepts modifications of the register in DEFAULT_CONFIG or CONFIG state only.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DSI1*	DSI0*
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	MAPO[4:0]*					—	—	APO[5:0]*					
Initial value:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
17	DSI1	0	R/W	Dynamic Slot Idle Phase (gdDynamicSlotIdlePhase) The duration of the dynamic slot idle phase has to be greater or equal than the idle detection time. Must be identical in all nodes of a cluster. Valid values are 0 to 2 Minislot.
16	DSI0	0	R/W	
15 to 13	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
12 to 8	MAPO[4:0]	H'01	R/W	Minislot Action Point Offset (gdMinislotActionPointOffset) Configures the action point offset in macroticks within the minislots of the dynamic segment. Must be identical in all nodes of a cluster. Valid values are 1 to 31 MT.
7, 6	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
5 to 0	APO[5:0]	H'01	R/W	Action Point Offset (gdActionPointOffset) Configures the action point offset in macroticks within static slots and symbol window. Must be identical in all nodes of a cluster. Valid values are 1 to 63 MT.

19.8.17 FlexRay GTU Configuration Register 10 (FRGTUC10)

The CC accepts modifications of the register in DEFAULT_CONFIG or CONFIG state only.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	MRC[10:0]*										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	MOC[13:0]*													
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
26 to 16	MRC[10:0]	H'002	R/W	Maximum Rate Correction (pRateCorrectionOut) Holds the maximum permitted rate correction value to be applied by the internal clock synchronization algorithm. The CC checks only the internal rate correction value against the maximum rate correction value (absolute value). Valid values are 2 to 1923 µT.
15, 14	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
13 to 0	MOC[13:0]	H'0005	R/W	Maximum Offset Correction (pOffsetCorrectionOut) Holds the maximum permitted offset correction value to be applied by the internal clock synchronization algorithm (absolute value). The CC checks only the internal offset correction value against the maximum offset correction value. Valid values are 5 to 15266 µT.

19.8.18 FlexRay GTU Configuration Register 11 (FRGTUC11)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	ERC[2:0]*			—	—	—	—	—	EOC[2:0]*		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	ERCC1	ERCC0	—	—	—	—	—	—	EOCC1	EOCC0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
26 to 24	ERC[2:0]	All 0	R/W	External Rate Correction (pExternRateCorrection) Holds the external rate correction value in microticks to be applied by the internal clock synchronization algorithm. The value is subtracted/added from/to the calculated rate correction value. The value is applied during NIT. May be modified in DEFAULT_CONFIG or CONFIG state only. Valid values are 0 to 7 μT.
23 to 19	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
18 to 16	EOC[2:0]	All 0	R/W	External Offset Correction (pExternOffsetCorrection) Holds the external offset correction value in microticks to be applied by the internal clock synchronization algorithm. The value is subtracted/added from/to the calculated offset correction value. The value is applied during NIT. May be modified in DEFAULT_CONFIG or CONFIG state only. Valid values are 0 to 7 μT.
15 to 10	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
9	ERCC1	0	R/W	External Rate Correction Control (vExternRateControl) By writing to ERCC[1:0] the external rate correction is enabled as specified below. Should be modified only outside NIT. 0X: No external rate correction 10: External rate correction value subtracted from calculated rate correction value 11: External rate correction value added to calculated rate correction value
8	ERCC0	0	R/W	
7 to 2	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
1	EOCC1	0	R/W	External Offset Correction Control (vExternOffsetControl) By writing to EOCC[1:0] the external offset correction is enabled as specified below. Should be modified only outside NIT. 0X: No external offset correction 10: External offset correction value subtracted from calculated offset correction value 11: External offset correction value added to calculated offset correction value
0	EOCC0	0	R/W	

19.9 CC Status Registers

During 8-/16-bit accesses to status variables coded with more than 8/16 bits, the variable might be updated by the CC between two accesses (non-atomic read accesses). The status vector may change faster than the Host can poll the status vector, depending on the peripheral bus clock frequency.

19.9.1 FlexRay CC Status Vector (FRCCSV)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	PSL[5:0]						RCA[4:0]					WSV[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	CSI	CSAI	CSNI	—	—	SLM1	SLM0	HRQ	FSI	POCS[5:0]					
Initial value:	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31, 30	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
29 to 24	PSL[5:0]	All 0	R	POC Status Log Status of POCS[5:0] immediately before entering HALT state. Set when entering HALT state. Set to HALT when FREEZE command is applied during HALT state and FSI is not already set i.e. the HALT state was not reached by FREEZE command. Reset to "B'000000" when leaving HALT state.
23 to 19	RCA[4:0]	H'02	R	Remaining Coldstart Attempts (vRemainingColdstartAttempts) Indicates the number of remaining coldstart attempts. The RUN command resets this counter to the maximum number of coldstart attempts as configured by FRSUCC1.CSA[4:0]. The initial value of RCA[4:0] during CONFIG and DEFAULT_CONFIG state is also FRSUCC1.CSA[4:0].

Bit	Bit Name	Initial Value	R/W	Description
18 to 16	WSV[2:0]	All 0	R	Wakeup Status (vPOC!WakeupStatus) Indicates the status of the current wakeup attempt. Reset by CHI command RESET_STATUS_INDICATORS or by transition from HALT to DEFAULT_CONFIG state or from READY to STARTUP state. 000: UNDEFINED. Wakeup not yet executed by the CC. 001: RECEIVED_HEADER. Set when the CC finishes wakeup due to the reception of a frame header without coding violation on either channel in WAKEUP_LISTEN state. 010: RECEIVED_WUP. Set when the CC finishes wakeup due to the reception of a valid wakeup pattern on the configured wakeup channel in WAKEUP_LISTEN state. 011: COLLISION_HEADER. Set when the CC stops wakeup due to a detected collision during wakeup pattern transmission by receiving a valid header on either channel. 100: COLLISION_WUP. Set when the CC stops wakeup due to a detected collision during wakeup pattern transmission by receiving a valid wakeup pattern on the configured wakeup channel. 101: COLLISION_UNKNOWN. Set when the CC stops wakeup by leaving WAKEUP_DETECT state after expiration of the wakeup timer without receiving a valid wakeup pattern or a valid frame header. 110: TRANSMITTED. Set when the CC has successfully completed the transmission of the wakeup pattern. 111: Reserved
15	—	0	R	Reserved. This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
14	CSI	1	R	Cold Start Inhibit (vColdStartInhibit) Indicates that the node is disabled from cold starting. The flag is set whenever the POC enters READY state due to CHI command READY. The flag has to be reset under control of the Host by CHI command ALLOW_COLDSTART (FRSUCC1.CMD[3:0] = "1001"). 0: Cold starting of node enabled 1: Cold starting of node disabled
13	CSAI	0	R	Coldstart Abort Indicator * Indicates that cold starting has been aborted. 0: No change 1: Coldstart aborted
12	CSNI	0	R	Coldstart Noise Indicator (vPOC!ColdstartNoise) * Indicates that the cold start procedure occurred under noisy conditions. 0: No change 1: Cold start procedure occurred under noisy conditions
11, 10	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
9	SLM1	0	R	Slot Mode (vPOC!SlotMode)
8	SLM0	0	R	Indicates the actual slot mode of the POC in states READY, WAKEUP, STARTUP, NORMAL_ACTIVE, and NORMAL_PASSIVE. Default is SINGLE. Changes to ALL, depending on FRSUCC1.TSM. In NORMAL_ACTIVE or NORMAL_PASSIVE state the CHI command ALL_SLOTS will change the slot mode from SINGLE over ALL_PENDING to ALL. Set to SINGLE in all other states. 00: SINGLE 01: Reserved 10: ALL_PENDING 11: ALL
7	HRQ	0	R	Halt Request (vPOC!CHIHaltRequest) Indicates that a request from the Host has been received to halt the POC at the end of the communication cycle. Reset by transition from HALT to DEFAULT_CONFIG state or when entering READY state. 0: No change 1: Transition to HALT state has been requested
6	FSI	0	R	Freeze Status Indicator (vPOC!Freeze) Indicates that the POC has entered the HALT state due to CHI command FREEZE or due to an error condition requiring an immediate POC halt. Reset by transition from HALT to DEFAULT_CONFIG state. 0: No change 1: HALT state entered due to FREEZE command or an error

Bit	Bit Name	Initial Value	R/W	Description
5 to 0	POCS[5:0]	All 0	R	Protocol Operation Control Status Indicates the actual state of operation of the CC Protocol Operation Control 00 0000: DEFAULT_CONFIG state 00 0001: READY state 00 0010: NORMAL_ACTIVE state 00 0011: NORMAL_PASSIVE state 00 0100: HALT state 00 0101...00 1110: Reserved 00 1111: CONFIG state Indicates the actual state of operation of the POC in the wakeup path 01 0000: WAKEUP_STANDBY state 01 0001: WAKEUP_LISTEN state 01 0010: WAKEUP_SEND state 01 0011: WAKEUP_DETECT state 01 0100...01 1111: Reserved Indicates the actual state of operation of the POC in the startup path 10 0000: STARTUP_PREPARE state 10 0001: COLDSTART_LISTEN state 10 0010: COLDSTART_COLLISION_RESOLUTION state 10 0011: COLDSTART_CONSISTENCY_CHECK state 10 0100: COLDSTART_GAP state 10 0101: COLDSTART_JOIN State 10 0110: INTEGRATION_COLDSTART_CHECK state 10 0111: INTEGRATION_LISTEN state 10 1000: INTEGRATION_CONSISTENCY_CHECK state 10 1001: INITIALIZE_SCHEDULE state 10 1010: ABORT_STARTUP state 10 1011: STARTUP_SUCCESS state 10 1100...11 1111: Reserved

Note: * Reset by CHI command RESET_STATUS_INDICATORS or by transition from HALT to DEFAULT_CONFIG state or from READY to STARTUP state.

19.9.2 FlexRay CC Error Vector (FRCCEV)

Reset by transition from HALT to DEFAULT_CONFIG state or when entering READY state.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	PTAC[4:0]					ERRM1	ERRM0	—	—	CCFC[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 13	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
12 to 8	PTAC[4:0]	All 0	R	Passive to Active Count (vAllowPassiveToActive) Indicates the number of consecutive even/odd cycle pairs that have passed with valid rate and offset correction terms, while the node is waiting to transit from NORMAL_PASSIVE state to NORMAL_ACTIVE state. The transition takes place when PTAC[4:0] equals FRSUCC1.PTA[4:0] – 1.
7	ERRM1	0	R	Error Mode (vPOC!ErrorMode) Indicates the actual error mode of the POC. 00: ACTIVE (green) 01: PASSIVE (yellow) 10: COMM_HALT (red) 11: Reserved
6	ERRM0	0	R	
5, 4	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
3 to 0	CCFC[3:0]	All 0	R	Clock Correction Failed Counter (vClockCorrectionFailed) The Clock Correction Failed Counter is incremented by one at the end of any odd communication cycle where either the missing offset correction error or missing rate correction error is active. The Clock Correction Failed Counter is reset to '0' at the end of an odd communication cycle if neither the offset correction failed nor the rate correction failed errors are active. The Clock Correction Failed Counter stops at 15.

19.9.3 FlexRay Slot Counter Value (FRSCV)

The register is reset when the CC leaves CONFIG state or enters STARTUP state.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	SCCB[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SCCA[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
26 to 16	SCCB[10:0]	All 0	R	Slot Counter Channel B (vSlotCounter[B]) Current slot counter value on channel B. The value is incremented by the CC and reset at the start of a communication cycle. Valid values are 0 to 2047.
15 to 11	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
10 to 0	SCCA[10:0]	All 0	R	Slot Counter Channel A (vSlotCounter[A]) Current slot counter value on channel A. The value is incremented by the CC and reset at the start of a communication cycle. Valid values are 0 to 2047.

19.9.4 FlexRay Macrotick and Cycle Counter Value (FRMTCCV)

The register is reset when the CC leaves CONFIG state or enters STARTUP state.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	CCV[5:0]					
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	MTV[13:0]													
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 22	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
21 to 16	CCV[5:0]	All 0	R	Cycle Counter Value (vCycleCounter) Current cycle counter value. The value is incremented by the CC at the start of a communication cycle. Valid values are 0 to 63.
15, 14	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
13 to 0	MTV[13:0]	All 0	R	Macrotick Value (vMacrotick) Current macrotick value. The value is incremented by the CC and reset at the start of a communication cycle. Valid values are 0 to 15999.

19.9.5 FlexRay Rate Correction Value (FRRCV)

The register is reset when the CC leaves CONFIG state or enters STARTUP state.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	RCV[11:0]											
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
11 to 0	RCV[11:0]	All 0	R	Rate Correction Value (vRateCorrection) Calculated internal rate correction value before limitation (two's complement). * If the RCV value exceeds the limits defined by FRGTUC10.MRC[10:0], flag FRSFS.RCLR is set to '1'.

Note: * The external rate correction value is added to the limited rate correction value.

19.9.6 FlexRay Offset Correction Value (FROCV)

The register is reset when the CC leaves CONFIG state or enters STARTUP state.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	OCV[18:16]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OCV[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 19	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
18 to 0	OCV[18:0]	All 0	R	Offset Correction Value (vOffsetCorrection) Calculated internal offset correction value before limitation (two's complement). * If the OCV value exceeds the limits defined by FRGTUC10.MOC[13:0], flag FRSFS.OCLR is set to '1'.

Note: * The external offset correction value is added to the limited rate correction value.

19.9.7 FlexRay Sync Frame Status (FRSFS)

The maximum number of valid sync frames in a communication cycle is 15.

The register is reset when the CC leaves CONFIG state or enters STARTUP state.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	RCLR	MRCS	OCLR	MOCS
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VSBO[3:0]				VSBE[3:0]				VSAO[3:0]				VSAE[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 20	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
19	RCLR	0	R	Rate Correction Limit Reached The Rate Correction Limit Reached flag signals to the Host, that the rate correction value has exceeded its limit as defined by FRGTUC10.MRC[10:0]. The flag is updated by the CC at start of offset correction phase. 0: Rate correction below limit 1: Rate correction limit reached
18	MRCS	0	R	Missing Rate Correction Signal The Missing Rate Correction flag signals to the Host, that no rate correction calculation can be performed because no pairs of even/odd sync frames were received. The flag is updated by the CC at start of offset correction phase. 0: Rate correction signal valid 1: Missing rate correction signal

Bit	Bit Name	Initial Value	R/W	Description
17	OCLR	0	R	Offset Correction Limit Reached The Offset Correction Limit Reached flag signals to the Host, that the offset correction value has exceeded its limit as defined by FRGTUC10.MOC[13:0]. The flag is updated by the CC at start of offset correction phase. 0: Offset correction below limit 1: Offset correction limit reached
16	MOCS	0	R	Missing Offset Correction Signal The Missing Offset Correction flag signals to the Host, that no offset correction calculation can be performed because no sync frames were received. The flag is updated by the CC at start of offset correction phase. 0: Offset correction signal valid 1: Missing offset correction signal
15 to 12	VSBO[3:0]	All 0	R	Valid Sync Frames Channel B, odd communication cycle *[1] *[2] Holds the number of valid sync frames received on channel B in the odd communication cycle. The value is updated during the NIT of each odd communication cycle.
11 to 8	VSBE[3:0]	All 0	R	Valid Sync Frames Channel B, even communication cycle *[1] *[2] Holds the number of valid sync frames received on channel B in the even communication cycle. The value is updated during the NIT of each even communication cycle.
7 to 4	VSAO[3:0]	All 0	R	Valid Sync Frames Channel A, odd communication cycle *[1] *[3] Holds the number of valid sync frames received on channel A in the odd communication cycle. The value is updated during the NIT of each odd communication cycle.

Bit	Bit Name	Initial Value	R/W	Description
3 to 0	VSAE[3:0]	All 0	R	Valid Sync Frames Channel A, even communication cycle *[1] *[3] Holds the number of valid sync frames received on channel A in the even communication cycle. The value is updated during the NIT of each even communication cycle.

Notes: 1. If transmission of sync frames is enabled by FRSUCC1.TXSY, the value is incremented by one.

2. This bit is only valid if FRSUCC1.CCHB is "1".

3. This bit is only valid if FRSUCC1.CCHA is "1".

19.9.8 FlexRay Symbol Window and NIT Status (FRSWNIT)

Symbol window related status information. Updated by the CC at the end of the symbol window for each channel. During startup the status data is not updated.

The register is reset when the CC leaves CONFIG state or enters STARTUP state.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	SBNB	SENB	SBNA	SENA	MTSB	MTSA	TCSB	SBSB	SESB	TCSA	SBSA	SESA
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
11	SBNB	0	R	Slot Boundary Violation during NIT Channel B (vSS!BViolationB) 0: No slot boundary violation detected 1: Slot boundary violation during NIT detected on channel B
10	SENB	0	R	Syntax Error during NIT Channel B (vSS!SyntaxErrorB) 0: No syntax error detected 1: Syntax error during NIT detected on channel B
9	SBNA	0	R	Slot Boundary Violation during NIT Channel A (vSS!BViolationA) 0: No slot boundary violation detected 1: Slot boundary violation during NIT detected on channel A
8	SENA	0	R	Syntax Error during NIT Channel A (vSS!SyntaxErrorA) 0: No syntax error detected 1: Syntax error during NIT detected on channel A

Bit	Bit Name	Initial Value	R/W	Description
7	MTSB	0	R	MTS Received on Channel B (vSS!ValidMTSB) Media Access Test symbol received on channel B during the preceding symbol window. Updated by the CC for each channel at the end of the symbol window. When this bit is set to '1', also interrupt flag FRSIR.MTSB is set to '1'. 0: No MTS symbol received on channel B 1: MTS symbol received on channel B
6	MTSA	0	R	MTS Received on Channel A (vSS!ValidMTSA) Media Access Test symbol received on channel A during the preceding symbol window. Updated by the CC for each channel at the end of the symbol window. When this bit is set to '1', also interrupt flag FRSIR.MTSA is set to '1'. 0: No MTS symbol received on channel A 1: MTS symbol received on channel A
5	TCSB	0	R	Transmission Conflict in Symbol Window Channel B (vSS!TxConflictB) 0: No transmission conflict detected 1: Transmission conflict in symbol window detected on channel B
4	SBSB	0	R	Slot Boundary Violation in Symbol Window Channel B (vSS!BViolationB) 0: No slot boundary violation detected 1: Slot boundary violation during symbol window detected on channel B
3	SESB	0	R	Syntax Error in Symbol Window Channel B (vSS!SyntaxErrorB) 0: No syntax error detected 1: Syntax error during symbol window detected on channel B
2	TCSA	0	R	Transmission Conflict in Symbol Window Channel A (vSS!TxConflictA) 0: No transmission conflict detected 1: Transmission conflict in symbol window detected on channel A

Bit	Bit Name	Initial Value	R/W	Description
1	SBSA	0	R	Slot Boundary Violation in Symbol Window Channel A (vSS!BViolationA) 0: No slot boundary violation detected 1: Slot boundary violation during symbol window detected on channel A
0	SESA	0	R	Syntax Error in Symbol Window Channel A (vSS!SyntaxErrorA) 0: No syntax error detected 1: Syntax error during symbol window detected on channel A

19.9.9 FlexRay Aggregated Channel Status (FRACS)

The aggregated channel status provides the Host with an accrued status of channel activity for all communication slots regardless of whether they are assigned for transmission or subscribed for reception. The aggregated channel status also includes status data from the symbol window and the network idle time. The status data is updated (set) after each slot and aggregated until it is reset by the Host. During startup the status data is not updated. A flag is cleared by writing a '1' to the corresponding bit position. Writing a '0' has no effect on the flag. A hardware reset or a software reset will also clear the register.

The register is reset when the CC leaves CONFIG state or enters STARTUP state.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	SBVB	CIB	CEDB	SEDB	VFRB	—	—	—	SBVA	CIA	CEDA	SEDA	VFRA
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only 1 can be written to clear the flag.

Bit	Bit Name	Initial Value	R/W	Description
31 to 13	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
12	SBVB	0	R/W	Slot Boundary Violation on Channel B (vSS!BViolationB) *[1] One or more slot boundary violations were observed on channel B at any time during the observation period (static or dynamic slots, symbol window, and NIT). 0: No slot boundary violation observed 1: Slot boundary violation(s) observed on channel B

Bit	Bit Name	Initial Value	R/W	Description
11	CIB	0	R/W	Communication Indicator Channel B *[1] *[2] One or more valid frames were received on channel B in slots that also contained any additional communication during the observation period, i.e. one or more slots received a valid frame AND had any combination of either syntax error, content error, OR slot boundary violation. 0: No valid frame(s) received in slots containing any additional communication 1: Valid frame(s) received on channel B in slots containing any additional communication
10	CEDB	0	R/W	Content Error Detected on Channel B (vSS!ContentErrorB) *[1] One or more frames with a content error were received on channel B in any static or dynamic slot during the observation period. 0: No frame with content error received 1: Frame(s) with content error received on channel B
9	SEDB	0	R/W	Syntax Error Detected on Channel B (vSS!SyntaxErrorB) *[1] One or more syntax errors in static or dynamic slots, symbol window, and NIT were observed on channel B. 0: No syntax error observed 1: Syntax error(s) observed on channel B
8	VFRB	0	R/W	Valid Frame Received on Channel B (vSS!ValidFrameB) One or more valid frames were received on channel B in any static or dynamic slot during the observation period. Reset under control of the Host. 0: No valid frame received 1: Valid frame(s) received on channel B
7 to 5	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
4	SBVA	0	R/W	Slot Boundary Violation on Channel A (vSS!BViolationA) *1 One or more slot boundary violations were observed on channel A at any time during the observation period (static or dynamic slots, symbol window, and NIT). 0: No slot boundary violation observed 1: Slot boundary violation(s) observed on channel A
3	CIA	0	R/W	Communication Indicator Channel A *1 *2 One or more valid frames were received on channel A in slots that also contained any additional communication during the observation period, i.e. one or more slots received a valid frame AND had any combination of either syntax error, content error, OR slot boundary violation. 0: No valid frame(s) received in slots containing any additional communication 1: Valid frame(s) received on channel A in slots containing any additional communication
2	CEDA	0	R/W	Content Error Detected on Channel A (vSS!ContentErrorA) *1 One or more frames with a content error were received on channel A in any static or dynamic slot during the observation period. 0: No frame with content error received 1: Frame(s) with content error received on channel A
1	SEDA	0	R/W	Syntax Error Detected on Channel A (vSS!SyntaxErrorA) *1 One or more syntax errors in static or dynamic slots, symbol window, and NIT were observed on channel A. 0: No syntax error observed 1: Syntax error(s) observed on channel A
0	VFRA	0	R/W	Valid Frame Received on Channel A (vSS!ValidFrameA) One or more valid frames were received on channel A in any static or dynamic slot during the observation period. 0: No valid frame received 1: Valid frame(s) received on channel A

Notes: 1. When one of the flags SEDB, CEDB, CIB, and SBVB changes from '0' to '1', interrupt flag FREIR.EDB is set to '1'. When one of the flags SEDA, CEDA, CIA, and SBVA changes from '0' to '1', interrupt flag FREIR.EDA is set to '1'.

2. The set condition of flags CIA and CIB is also fulfilled if there is only one single frame in the slot and the slot boundary at the end of the slot is reached during the frames channel idle recognition phase.

19.9.10 FlexRay Even Sync ID [1…15] (FRESIDn)

Registers FRESID1 to FRESID15 hold the frame IDs of the sync frames received in even communication cycles used for clock synchronization up to the limit of gSyncNodeMax. The values are sorted in ascending order, with register FRESID1 holding the lowest received sync frame ID. If the node itself transmits a sync frame in an even communication cycle, register FRESID1 holds the respective sync frame ID as configured in message buffer 0 and flags RXEA, RXEB are set. The value is updated during the NIT of each even communication cycle.

The register is reset when the CC leaves CONFIG state or enters STARTUP state.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RXEB	RXEA	—	—	—	—	EID[9:0]									
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
15	RXEB	0	R	Received/Configured Even Sync ID on Channel B Signals that a sync frame corresponding to the stored even sync ID was received on channel B or that the node is configured to be a sync node with key slot = EID[9:0] (FRESID1 only). 0: No sync frame received on channel B/node not configured to transmit sync frames 1: Sync frame received on channel B/node configured to transmit sync frames

Bit	Bit Name	Initial Value	R/W	Description
14	RXEA	0	R	Received/Configured Even Sync ID on Channel A Signals that a sync frame corresponding to the stored even sync ID was received on channel A or that the node is configured to be a sync node with key slot = EID[9:0] (FRESID1 only). 0: No sync frame received on channel A/node not configured to transmit sync frames 1: Sync frame received on channel A/node configured to transmit sync frames
13 to 10	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
9 to 0	EID[9:0]	All 0	R	Even Sync ID (vsSyncIDListA,B even) Sync frame ID even communication cycle.

19.9.11 FlexRay Odd Sync ID [1…15] (FROSIDn)

Registers FROSID1 to FROSID15 hold the frame IDs of the sync frames received in odd communication cycles used for clock synchronization up to the limit of gSyncNodeMax. The values are sorted in ascending order, with register FROSID1 holding the lowest received sync frame ID. If the node itself transmits a sync frame in an odd communication cycle, register FROSID1 holds the respective sync frame ID as configured in message buffer 0 and flags RXOA, RXOB are set. The value is updated during the NIT of each odd communication cycle.

The register is reset when the CC leaves CONFIG state or enters STARTUP state.

Bit:	31	30	29	28	27	26	25	24	23	22	22	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RXOB	RXOA	—	—	—	—	OID[9:0]									
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
15	RXOB	0	R	Received/Configured Odd Sync ID on Channel B Signals that a sync frame corresponding to the stored odd sync ID was received on channel B or that the node is configured to be a sync node with key slot = OID[9:0] (FROSID1 only). 0: No sync frame received on channel B/node not configured to transmit sync frames 1: Sync frame received on channel B/node configured to transmit sync frames

Bit	Bit Name	Initial Value	R/W	Description
14	RXOA	0	R	Received/Configured Odd Sync ID on Channel A Signals that a sync frame corresponding to the stored odd sync ID was received on channel A or that the node is configured to be a sync node with key slot = OID[9:0] (FROSID1 only). 0: No sync frame received on channel A/node not configured to transmit sync frames 1: Sync frame received on channel A/node configured to transmit sync frames
13 to 10	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
9 to 0	OID[9:0]	All 0	R	Odd Sync ID (vsSyncIDListA,B odd) Sync frame ID odd communication cycle.

19.9.12 FlexRay Network Management Vector [1…3] (FRNMVn)

The three network management registers hold the accrued NM vector (configurable 0 to 12 bytes). The accrued NM vector is generated by the CC by bit-wise ORing each NM vector received (valid static frames with PPI = '1') on each channel (see section 19.19, Network Management).

The CC updates the NM vector at the end of each communication cycle as long as the CC is either in NORMAL_ACTIVE or NORMAL_PASSIVE state.

FRNMVn-bytes exceeding the configured NM vector length are not valid.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	NM[31:24]								NM[23:16]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NM[15:8]								NM[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	NM[31:24]	All 0	R	Hold the accrued NM vector (network management vector). The accrued NM vector is generated by bit-wise ORing each NM vector received on each channel. An NM vector is 0 to 12 bytes long.
23 to 16	NM[23:16]	All 0	R	
15 to 8	NM[15:8]	All 0	R	
7 to 0	NM[7:0]	All 0	R	

Table 19.4 shows the assignment of the received payload's data bytes to the network management vector.

Table 19.4 (1) Assignment of Data Bytes to Network Management Vector (When Byte Swap is Disabled: FXROC.FEC = 0)

	NM31 to NM24	NM23 to NM16	NM15 to NM8	NM7 to NM0
FRNMV1	Data3	Data2	Data1	Data0
FRNMV2	Data7	Data6	Data5	Data4
FRNMV3	Data11	Data10	Data9	Data8

Table 19.4 (2) Assignment of Data Bytes to Network Management Vector (When Byte Swap is Enabled: FXROC.FEC = 1)

	NM31 to NM24	NM23 to NM16	NM15 to NM8	NM7 to NM0
FRNMV1	Data0	Data1	Data2	Data3
FRNMV2	Data4	Data5	Data6	Data7
FRNMV3	Data8	Data9	Data10	Data11

19.10 Message Buffer Control Registers

19.10.1 FlexRay Message RAM Configuration (FRMRC)

The FlexRay Message RAM Configuration register defines the number of message buffers assigned to the static segment, dynamic segment, and FIFO. The register can be written during DEFAULT_CONFIG or CONFIG state only.

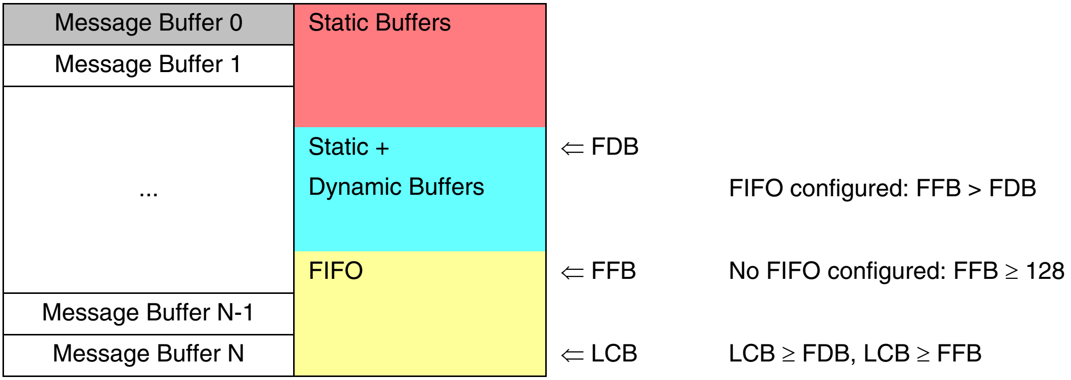
Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	SPLM	SEC1	SEC0	LCB[7:0]							
Initial value:	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FFB[7:0]								FDB[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
26	SPLM	0	R/W	Sync Frame Payload Multiplex This bit is only evaluated if the node is configured as sync node (FRSUCC1.TXSY = '1') or for single slot mode operation (FRSUCC1.TSM = '1'). When this bit is set to '1' message buffers 0 and 1 are dedicated for sync frame transmission with different payload data on channels A and B. When this bit is set to '0', sync frames are transmitted from message buffer 0 with the same payload data on both channels. Note that the channel filter configuration for message buffer 0 resp. message buffer 1 has to be chosen accordingly. 0: Only message buffer 0 locked against reconfiguration 1: Both message buffers 0 and 1 are locked against reconfiguration Note: In case the node is configured as sync node (FRSUCC1.TXSY = '1') or for single slot mode operation (FRSUCC1.TSM = '1'), message buffer 0 resp. 1 is reserved for sync frames or single slot frames and have to be configured with the node-specific key slot ID. In case the node is neither configured as sync node nor for single slot operation message buffer 0 resp. 1 is treated like all other message buffers.

Bit	Bit Name	Initial Value	R/W	Description
25	SEC1	0	R/W	Secure Buffers Not evaluated when the CC is in DEFAULT_CONFIG or CONFIG state. 00: Reconfiguration of message buffers enabled with numbers < FFB enabled Note: In nodes configured for sync frame transmission or for single slot mode operation, message buffer 0 (and if SPLM = '1', also message buffer 1) is always locked 01: Reconfiguration of message buffers with numbers < FDB and with numbers ≥ FFB locked and transmission of message buffers for static segment with numbers ≥ FDB disabled 10: Reconfiguration of all message buffers locked 11: Reconfiguration of all message buffers locked and transmission of message buffers for static segment with numbers ≥ FDB disabled
24	SEC0	1	R/W	
23 to 16	LCB[7:0]	H'80	R/W	Last Configured Buffer 0...127: Number of message buffers is LCB + 1 ≥128: No message buffer configured
15 to 8	FFB[7:0]	All 0	R/W	First Buffer of FIFO 0: All message buffers assigned to the FIFO 1...127: Message buffers from FFB to LCB assigned to the FIFO ≥128: No message buffer assigned to the FIFO
7 to 0	FDB[7:0]	All 0	R/W	First Dynamic Buffer 0: No group of message buffers exclusively for the static segment configured 1...127: Message buffers 0 to FDB – 1 reserved for static segment ≥128: No dynamic message buffers configured



The programmer has to ensure that the configuration defined by FDB[7:0], FFB[7:0], and LCB[7:0] is valid. The CC does not check for erroneous configurations.

Note: The maximum number of header sections is 128. This means a maximum of 128 message buffers can be configured. The maximum length of a data section is 254 bytes. The length of the data section may be configured differently for each message buffer. For details see section 19.25, Message RAM.

In case two or more message buffers are assigned to slot 1 by use of cycle filtering, all of them must be located either in the "Static Buffers" or at the beginning of the "Static + Dynamic Buffers" section.

The FlexRay protocol specification requires that each node have to send a frame in its key slot. Therefore at least message buffer 0 is reserved for transmission in the key slot. Due to this requirement a maximum number of 127 message buffers can be assigned to the FIFO. Nevertheless, a non protocol conformant configuration without a transmission slot in the static segment would still be operational.

The payload length configured and the length of the data section need to be configured identical for all message buffers belonging to the FIFO via FRWRHS2.PLC[6:0] and FRWRHS3.DP[10:0]. When the CC is not in DEFAULT_CONFIG or CONFIG state reconfiguration of message buffers belonging to the FIFO is locked.

19.10.2 FlexRay FIFO Rejection Filter (FRFRF)

The FlexRay FIFO Rejection Filter defines a user specified sequence of bits to which channel, frame ID, and cycle count of the incoming frames are compared. Together with the FlexRay FIFO Rejection Filter Mask this register determines whether a message is rejected by the FIFO. The FRFRF register can be written during DEFAULT_CONFIG or CONFIG state only.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	RNF	RSS	CYF[6:0]						
Initial value:	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	FID[10:0]											CH1	CH0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 25	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
24	RNF	1	R/W	Reject Null Frames If this bit is set, received null frames are not stored in the FIFO. 0: Null frames are stored in the FIFO 1: Reject all null frames
23	RSS	1	R/W	Reject in Static Segment If this bit is set, the FIFO is used only for the dynamic segment. 0: FIFO also used for static segment 1: Reject messages in static segment
22 to 16	CYF[6:0]	All 0	R/W	Cycle Counter Filter The 7-bit cycle counter filter determines the cycle set to which frame ID and channel rejection filter are applied. In cycles not belonging to the cycle set specified by CYF[6:0], all frames are rejected. For details about the configuration of the cycle counter filter see section 19.20.2, Cycle Counter Filtering.

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
12 to 2	FID[10:0]	All 0	R/W	Frame ID Filter Determines the frame ID to be rejected by the FIFO. With the additional configuration of register FRFRFM, the corresponding frame ID filter bits are ignored, which results in further rejected frame IDs. When FRFRFM.MFID[10:0] is zero, a frame ID filter value of zero means that no frame ID is rejected. 0...2047 = Frame ID filter values
1	CH1	0	R/W	Channel Filter
0	CH0	0	R/W	00: Receive on both channels 01: Receive only on channel B 10: Receive only on channel A 11: No reception Note: If reception on both channels is configured, also in static segment always both frames (from channel A and B) are stored in the FIFO, even if they are identical.

19.10.3 FlexRay FIFO Rejection Filter Mask (FRFRFM)

The FlexRay FIFO Rejection Filter Mask specifies which of the corresponding frame ID filter bits are relevant for rejection filtering. If a bit is set, it indicates that the corresponding bit in the FRFRF register will not be considered for rejection filtering. The FRFRFM register can be written during DEFAULT_CONFIG or CONFIG state only.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	MFID[10:0]											—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 13	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
12 to 2	MFID[10:0]	All 0	R/W	Mask Frame ID Filter 0: Corresponding frame ID filter bit is used for rejection filtering 1: Ignore corresponding frame ID filter bit.
1, 0	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.

19.10.4 FlexRay FIFO Critical Level (FRFCL)

The CC accepts modifications of the register in DEFAULT_CONFIG or CONFIG state only.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CL[7:0]							
Initial value:	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
7 to 0	CL[7:0]	H'80	R/W	Critical Level When the receive FIFO fill level FRFSR.RFFL[7:0] is equal or greater than the critical level configured by CL[7:0], the receive FIFO critical level flag FRFSR.RFCL is set. If CL[7:0] is programmed to values > 128, bit FRFSR.RFCL is never set. When FRFSR.RFCL changes from '0' to '1' bit FRSIR.RFCL is set to '1', and if enabled, an interrupt is generated.

19.11 Message Buffer Status Registers

19.11.1 FlexRay Message Handler Status (FRMHDS)

A flag is cleared by writing a '1' to the corresponding bit position. Writing a '0' has no effect on the flag. The register will also be cleared by a hardware reset, a software reset, or CHI command CLEAR_RAMS.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	MBU[6:0]							—	MBT[6:0]						
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	FMB[6:0]							CRAM	MFMB	FMBD	PTBF2	PTBF1	PMR	POBF	PIBF
Initial value:	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only 1 can be written to clear the flag.

Bit	Bit Name	Initial Value	R/W	Description
31	—	0	R	Reserved. This bit is always read as 0. The write value should always be 0.
30 to 24	MBU[6:0]	All 0	R	Message Buffer Updated *1 Number of message buffer that was updated last by the CC. For this message buffer the respective ND and/or MBC flag in the FRNDAT1/2/3/4 registers and the FRMBSC1/2/3/4 registers are also set.
23	—	0	R	Reserved. This bit is always read as 0. The write value should always be 0.
22 to 16	MBT[6:0]	All 0	R	Message Buffer Transmitted *1 Number of last successfully transmitted message buffer. If the message buffer is configured for single-shot mode, the respective TXR flag in the FRTXRQ1/2/3/4 registers was reset.
15	—	0	R	Reserved. This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
14 to 8	FMB[6:0]	All 0	R	Faulty Message Buffer Parity error occurred when reading from the message buffer or when transferring data from Input Buffer or Transient Buffer 1,2 to the message buffer referenced by FMB[6:0]. Value only valid when one of the flags PIBF, PMR, PTBF1, PTBF2, and flag FMBD is set. Updated only after the Host has reset flag FMBD. Is not updated while flag FMBD is set to 1.
7	CRAM	1	R	Clear all internal RAM's Signals that execution of the CHI command CLEAR_RAMS is ongoing (all bits of all internal RAM blocks are written to '0'). The bit is set by hardware or software reset or by CHI command CLEAR_RAMS. 0: No execution of the CHI command CLEAR_RAMS 1: Execution of the CHI command CLEAR_RAMS ongoing
6	MFMB	0	R/W	Multiple Faulty Message Buffers detected 0: No additional faulty message buffer 1: Another faulty message buffer was detected while flag FMBD is set
5	FMBD	0	R/W	Faulty Message Buffer Detected 0: No faulty message buffer 1: Message buffer referenced by FMB[6:0] holds faulty data due to a parity error
4	PTBF2	0	R/W	Parity Error Transient Buffer RAM B *2 0: No parity error 1: Parity error occurred when reading Transient Buffer RAM B
3	PTBF1	0	R/W	Parity Error Transient Buffer RAM A *2 0: No parity error 1: Parity error occurred when reading Transient Buffer RAM A
2	PMR	0	R/W	Parity Error Message RAM *2 0: No parity error 1: Parity error occurred when reading the Message RAM

Bit	Bit Name	Initial Value	R/W	Description
1	POBF	0	R/W	Parity Error Output Buffer RAM 1,2 *[2] 0: No parity error 1: Parity error occurred when reading Output Buffer RAM 1,2
0	PIBF	0	R/W	Parity Error Input Buffer RAM 1,2 *[2] 0: No parity error 1: Parity error occurred when reading Input Buffer RAM 1,2

Notes: 1. These flags are cleared to 0 when the CC leaves CONFIG state or enters STARTUP state.

2. When one of the flags PTBF2, PTBF1, PMR, POBF, and PIBF changes from '0' to '1', FREIR.PERR is set to '1'.

19.11.2 FlexRay Last Dynamic Transmit Slot (FRLDTS)

The register is reset when the CC leaves CONFIG state or enters STARTUP state.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	LDTB[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	LDTA[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
26 to 16	LDTB[10:0]	All 0	R	Last Dynamic Transmission Channel B Slot counter value at the time of the last frame transmission on channel B in the dynamic segment of this node. It is updated at the end of the dynamic segment and is reset to zero if no frame was transmitted during the dynamic segment.
15 to 11	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
10 to 0	LDTA[10:0]	All 0	R	Last Dynamic Transmission Channel A Slot counter value at the time of the last frame transmission on channel A in the dynamic segment of this node. It is updated at the end of the dynamic segment and is reset to zero if no frame was transmitted during the dynamic segment.

19.11.3 FlexRay FIFO Status Register (FRFSR)

The register is reset when the CC leaves CONFIG state or enters STARTUP state.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFFL[7:0]								—	—	—	—	—	RFO	RFCL	RFNE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
15 to 8	RFFL[7:0]	All 0	R	Receive FIFO Fill Level Number of FIFO buffers filled with new data not yet read by the Host. Maximum value is 128.
7 to 3	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
2	RFO	0	R	Receive FIFO Overrun The flag is set by the CC when a receive FIFO overrun is detected. When a receive FIFO overrun occurs, the oldest message is overwritten with the actual received message. In addition, interrupt flag FREIR.RFO is set. The flag is cleared by the next FIFO read access issued by the Host. 0: No receive FIFO overrun detected 1: A receive FIFO overrun has been detected

Bit	Bit Name	Initial Value	R/W	Description
1	RFCL	0	R	Receive FIFO Critical Level This flag is set when the receive FIFO fill level RFFL[7:0] is equal or greater than the critical level as configured by FRFCL.CL[7:0]. The flag is cleared by the CC as soon as RFFL[7:0] drops below FRFCL.CL[7:0].-+ When RFCL changes from '0' to '1' bit FRSIR.RFCL is set to '1', and if enabled, an interrupt is generated. 0: Receive FIFO below critical level 1: Receive FIFO critical level reached
0	RFNE	0	R	Receive FIFO Not Empty This flag is set by the CC when a received valid frame (data or null frame depending on rejection mask) was stored in the FIFO. In addition, interrupt flag FRSIR.RFNE is set. The bit is reset after the Host has read all messages from the FIFO. 0: Receive FIFO is empty 1: Receive FIFO is not empty

19.11.4 FlexRay Message Handler Constraints Flags (FRMHDF)

Some constraints exist for the Message Handler regarding the peripheral bus clock frequency, Message RAM configuration, and FlexRay bus traffic. To simplify software development, constraints violations are reported by setting flags in the FRMHDF.

A flag is cleared by writing a '1' to the corresponding bit position. Writing a '0' has no effect on the flag. A hardware reset or a software reset will also clear the register. The register is reset when the CC leaves CONFIG state or enters STARTUP state.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	WAHP	—	—	TBFB	TBFA	FNFB	FNFA	SNUB	SNUA
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/(W)*	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only 1 can be written to clear the flag.

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
8	WAHP	0	R/W	Write Attempt to Header Partition * Outside DEFAULT_CONFIG and CONFIG state this flag is set by the CC when the message handler tries to write message data into the header partition of the Message RAM due to faulty configuration of a message buffer. The write attempt is not executed, to protect the header partition from unintended write accesses. 0: No write attempt to header partition 1: Write attempt to header partition
7, 6	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
5	TBFB	0	R/W	Transient Buffer Access Failure B * This flag is set by the CC when a read or write access to TBF B requested by PRT B could not complete within the available time. 0: No TBF B access failure 1: TBF B access failure
4	TBFA	0	R/W	Transient Buffer Access Failure A * This flag is set by the CC when a read or write access to TBF A requested by PRT A could not complete within the available time. 0: No TBF A access failure 1: TBF A access failure
3	FNFB	0	R/W	Find Sequence Not Finished Channel B * This flag is set by the CC when the Message Handler, due to overload condition, was not able to finish a find sequence (scan of Message RAM for matching message buffer) with respect to channel B. 0: No find sequence not finished for channel B 1: Find sequence not finished for channel B
2	FNFA	0	R/W	Find Sequence Not Finished Channel A * This flag is set by the CC when the Message Handler, due to overload condition, was not able to finish a find sequence (scan of Message RAM for matching message buffer) with respect to channel A. 0: No find sequence not finished for channel A 1: Find sequence not finished for channel A
1	SNUB	0	R/W	Status Not Updated Channel B * This flag is set by the CC when the Message Handler, due to overload condition, was not able to update a message buffer's status FRMBS with respect to channel B. 0: No overload condition occurred when updating FRMBS for channel B 1: FRMBS for channel B not updated

Bit	Bit Name	Initial Value	R/W	Description
0	SNUA	0	R/W	Status Not Updated Channel A * This flag is set by the CC when the Message Handler, due to overload condition, was not able to update a message buffer's status FRMBS with respect to channel A. 0: No overload condition occurred when updating FRMBS for channel A 1: FRMBS for channel A not updated

Note: * When one of the flags SNUA, SNUB, FNFA, FNFB, TBFA, TBFB, and WAHP changes from '0' to '1', interrupt flag FREIR.MHF is set to '1'.

19.11.5 FlexRay Transmission Request 1/2/3/4 (FRTXRQ1/2/3/4)

The four registers reflect the state of the TXR flags of all configured message buffers. The flags are evaluated for transmit buffers only. If the number of configured message buffers is less than 128, the remaining TXR flags have no meaning.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TXR[127:112]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXR[111:96]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TXR[127:96]	All 0	R	Transmission Request If the flag is set, the respective message buffer is ready for transmission or transmission of this message buffer is in progress. In single-shot mode the flags are reset after transmission has completed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TXR[95:80]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXR[79:64]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TXR[95:64]	All 0	R	Transmission Request If the flag is set, the respective message buffer is ready for transmission or transmission of this message buffer is in progress. In single-shot mode the flags are reset after transmission has completed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TXR[63:48]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXR[47:32]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TXR[63:32]	All 0	R	Transmission Request If the flag is set, the respective message buffer is ready for transmission or transmission of this message buffer is in progress. In single-shot mode the flags are reset after transmission has completed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TXR[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXR[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TXR[31:0]	All 0	R	Transmission Request If the flag is set, the respective message buffer is ready for transmission or transmission of this message buffer is in progress. In single-shot mode the flags are reset after transmission has completed.

19.11.6 FlexRay New Data 1/2/3/4 (FRNDAT1/2/3/4)

The four registers reflect the state of the ND flags of all configured message buffers. ND flags belonging to transmit buffers have no meaning. If the number of configured message buffers is less than 128, the remaining ND flags have no meaning. The registers are reset when the CC leaves CONFIG state or enters STARTUP state.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ND[127:112]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ND[111:96]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	ND[127:96]	All 0	R	New Data The flags are set when a valid received data frame matches the message buffer's filter configuration, independent of the payload length received or the payload length configured for that message buffer. The flags are not set after reception of null frames except for message buffers belonging to the receive FIFO. An ND flag is reset when the header section of the corresponding message buffer is reconfigured or when the data section has been transferred to the Output Buffer.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ND[95:80]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ND[79:64]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	ND[95:64]	All 0	R	New Data The flags are set when a valid received data frame matches the message buffer's filter configuration, independent of the payload length received or the payload length configured for that message buffer. The flags are not set after reception of null frames except for message buffers belonging to the receive FIFO. An ND flag is reset when the header section of the corresponding message buffer is reconfigured or when the data section has been transferred to the Output Buffer.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ND[63:48]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ND[47:32]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	ND[63:32]	All 0	R	New Data The flags are set when a valid received data frame matches the message buffer's filter configuration, independent of the payload length received or the payload length configured for that message buffer. The flags are not set after reception of null frames except for message buffers belonging to the receive FIFO. An ND flag is reset when the header section of the corresponding message buffer is reconfigured or when the data section has been transferred to the Output Buffer.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ND[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ND[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	ND[31:0]	All 0	R	New Data The flags are set when a valid received data frame matches the message buffer's filter configuration, independent of the payload length received or the payload length configured for that message buffer. The flags are not set after reception of null frames except for message buffers belonging to the receive FIFO. An ND flag is reset when the header section of the corresponding message buffer is reconfigured or when the data section has been transferred to the Output Buffer.

19.11.7 FlexRay Message Buffer Status Changed 1/2/3/4 (FRMBSC1/2/3/4)

The four registers reflect the state of the MBC flags of all configured message buffers. If the number of configured message buffers is less than 128, the remaining MBC flags have no meaning. The registers are reset when the CC leaves CONFIG state or enters STARTUP state.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MBC[127:112]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBC[111:96]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	MBC[127:96]	All 0	R	Message Buffer Status Changed An MBC flag is set whenever the Message Handler changes one of the status flags VFRA, VFRB, SEOA, SEOB, CEOA, CEOB, SVOA, SVOB, TCIA, TCIB, ESA, ESB, MLST, FTA, and FTB in the header section (see section 19.13.5, FlexRay Message Buffer Status (FRMBS), and the Message Buffer Status description in section 19.25.1, Header Partition) of the respective message buffer. An MBC flag is reset when the header section of the corresponding message buffer is reconfigured or when it has been transferred to the Output Buffer.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MBC[95:80]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBC[79:64]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	MBC[95:64]	All 0	R	Message Buffer Status Changed An MBC flag is set whenever the Message Handler changes one of the status flags VFRA, VFRB, SEOA, SEOB, CEOA, CEOB, SVOA, SVOB, TCIA, TCIB, ESA, ESB, MLST, FTA, and FTB in the header section (see section 19.13.5, FlexRay Message Buffer Status (FRMBS), and the Message Buffer Status description in section 19.25.1, Header Partition) of the respective message buffer. An MBC flag is reset when the header section of the corresponding message buffer is reconfigured or when it has been transferred to the Output Buffer.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MBC[63:48]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBC[47:32]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	MBC[63:32]	All 0	R	Message Buffer Status Changed An MBC flag is set whenever the Message Handler changes one of the status flags VFRA, VFRB, SEOA, SEOB, CEOA, CEOB, SVOA, SVOB, TCIA, TCIB, ESA, ESB, MLST, FTA, and FTB in the header section (see section 19.13.5, FlexRay Message Buffer Status (FRMBS), and the Message Buffer Status description in section 19.25.1, Header Partition) of the respective message buffer. An MBC flag is reset when the header section of the corresponding message buffer is reconfigured or when it has been transferred to the Output Buffer.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MBC[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBC[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	MBC[31:0]	All 0	R	Message Buffer Status Changed An MBC flag is set whenever the Message Handler changes one of the status flags VFRA, VFRB, SEOA, SEOB, CEOA, CEOB, SVOA, SVOB, TCIA, TCIB, ESA, ESB, MLST, FTA, and FTB in the header section (see section 19.13.5, FlexRay Message Buffer Status (FRMBS), and the Message Buffer Status description in section 19.25.1, Header Partition) of the respective message buffer. An MBC flag is reset when the header section of the corresponding message buffer is reconfigured or when it has been transferred to the Output Buffer.

19.12 Input Buffer (IBF)

Double buffer structure consisting of Input Buffer Host and Input Buffer Shadow. While the Host can write to Input Buffer Host, the transfer to the Message RAM is done from Input Buffer Shadow. The Input Buffer holds the header and data sections to be transferred to the selected message buffer in the Message RAM. It is used to configure the message buffers in the Message RAM and to update the data sections of transmit buffers.

When updating the header section of a message buffer in the Message RAM from the Input Buffer, the FlexRay Message Buffer Status as described in section 19.13.5, FlexRay Message Buffer Status (FRMBS), is automatically reset to zero.

The header sections of message buffers belonging to the receive FIFO can only be (re)configured when the CC is in DEFAULT_CONFIG or CONFIG state. For those message buffers only the payload length configured and the data pointer need to be configured via FRWRHS2.PLC[6.0] and FRWRHS3.DP[10:0]. All information required for acceptance filtering is taken from the FlexRay FIFO rejection filter and the FlexRay FIFO rejection filter mask.

The data transfer between Input Buffer (IBF) and Message RAM is described in details in section 19.24.2 (1), Data Transfer from Input Buffer to Message RAM.

19.12.1 FlexRay Write Data Section [1…64] (FRWRDSn)

Holds the data words to be transferred to the data section of the addressed message buffer. The data words (DWn) are written to the Message RAM in transmission order from DW1 (byte0, byte1) to DWPL (PL = number of data words as defined by the payload length configured FRWRHS2.PLC[6:0]).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MD[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MD[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	MD[31:0]	All 0	R/W	Message Data (when byte swap is disabled: FXROC.FEC = 0) MD[7:0] = $DW_{2n - 1}$, $byte_{4n - 4}$ MD[15:8] = $DW_{2n - 1}$, $byte_{4n - 3}$ MD[23:16] = DW_{2n}, $byte_{4n - 2}$ MD[31:24] = DW_{2n}, $byte_{4n - 1}$ Message Data (when byte swap is enabled: FXROC.FEC = 1) MD[7:0] = DW_{2n}, $byte_{4n-1}$ MD[15:8] = DW_{2n}, $byte_{4n-2}$ MD[23:16] = $DW_{2n - 1}$, $byte_{4n - 3}$ MD[31:24] = $DW_{2n - 1}$, $byte_{4n - 4}$

Note: DW127 is located on FRWRDS64.MD[15:0]. In this case FRWRDS64.MD[31:16] is unused (no valid data). The Input Buffer RAMs are initialized to zero when leaving hardware reset or software reset, or by CHI command CLEAR_RAMS.
In the FlexRay bus, transmission is by the following order; WRDSn[7:0], WRDSn[15:8], WRDSn[23:16], WRDSn[31:24], and the highest order of bits will be transmitted first.

19.12.2 FlexRay Write Header Section 1 (FRWRHS1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	MBI	TXM	PPIT	CFG	CHB	CHA	—	CYC[6:0]						
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	FID[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31, 30	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
29	MBI	0	R/W	Message Buffer Interrupt This bit enables the receive/transmit interrupt for the corresponding message buffer. After a dedicated receive buffer has been updated by the Message Handler, flag FRSIR.RXI and/or FRSIR.MBSI are set. After a transmission has completed flag FRSIR.TXI is set. 0: The corresponding message buffer interrupt is disabled 1: The corresponding message buffer interrupt is enabled
28	TXM	0	R/W	Transmission Mode 0: Continuous mode 1: Single-shot mode This bit is used to select the transmission mode (see section 19.21.3, Transmit Buffers).

Bit	Bit Name	Initial Value	R/W	Description
27	PPIT	0	R/W	Payload Preamble Indicator Transmit This bit is used to control the state of the Payload Preamble Indicator in transmit frames. If the bit is set in a static message buffer, the respective message buffer holds network management information. If the bit is set in a dynamic message buffer the first two bytes of the payload segment may be used for message ID filtering by the receiver. Message ID filtering of received FlexRay frames is not supported by the FlexRay module, but can be done by the Host. 0: Payload Preamble Indicator not set 1: Payload Preamble Indicator set
26	CFG	0	R/W	Message Buffer Direction Configuration Bit This bit is used to configure the corresponding buffer as transmit buffer or as receive buffer. For message buffers belonging to the receive FIFO the bit is not evaluated. 0: The corresponding buffer is configured as Receive Buffer 1: The corresponding buffer is configured as Transmit Buffer

<table>
<tr><th>Bit</th><th>Bit Name</th><th>Initial Value</th><th>R/W</th><th>Description</th></tr>
<tr><td>25</td><td>CHB</td><td>0</td><td>R/W</td><td rowspan="2">Channel Filter Control Bits*
Transmit buffer:
<table>
<tr><th>CHB</th><th>CHA</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td>No transmission</td></tr>
<tr><td>0</td><td>1</td><td>Frame transmission on channel A</td></tr>
<tr><td>1</td><td>0</td><td>Frame transmission on channel B</td></tr>
<tr><td>1</td><td>1</td><td>Frame transmission on both channels (static segment only)</td></tr>
</table>
Receive buffer:
<table>
<tr><th>CHB</th><th>CHA</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td>Receive frame ignored</td></tr>
<tr><td>0</td><td>1</td><td>Store the frame received on channel A in the message buffer</td></tr>
<tr><td>1</td><td>0</td><td>Store the frame received on channel B in the message buffer</td></tr>
<tr><td>1</td><td>1</td><td>Store the first valid frame received on channels A or B in the message buffer (static segment only)</td></tr>
</table></td></tr>
<tr><td>24</td><td>CHA</td><td>0</td><td>R/W</td></tr>
<tr><td>23</td><td>—</td><td>0</td><td>R</td><td>Reserved.
This bit is always read as 0. The write value should always be 0.</td></tr>
<tr><td>22 to 16</td><td>CYC[6:0]</td><td>All 0</td><td>R/W</td><td>Cycle Code
The 7-bit cycle code determines the cycle set used for cycle counter filtering. For details about the configuration of the cycle code see section 19.20.2, Cycle Counter Filtering.</td></tr>
<tr><td>15 to 11</td><td>—</td><td>All 0</td><td>R</td><td>Reserved.
These bits are always read as 0. The write value should always be 0.</td></tr>
<tr><td>10 to 0</td><td>FID[10:0]</td><td>All 0</td><td>R/W</td><td>Frame ID
Frame ID of the selected message buffer.
The frame ID defines the slot number for transmission/reception of the respective message. Message buffers with frame ID = '0' are considered as not valid.</td></tr>
</table>

Note: * If a message buffer is configured for the dynamic segment and both bits of the channel filtering field are set to '1', no frames are transmitted resp. received frames are ignored (same function as CHA = CHB = '0')

19.12.3 FlexRay Write Header Section 2 (FRWRHS2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	PLC[6:0]						
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	CRC[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 23	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
22 to 16	PLC[6:0]	All 0	R/W	Payload Length Configured Length of data section (number of 2-byte words) as configured by the Host. During static segment the static frame payload length as configured by FRMHDC.SFDL[6:0] defines the payload length for all static frames. If the payload length configured by PLC[6:0] is shorter than this value padding bytes are inserted to ensure that frames have proper physical length. The padding pattern is logical zero (see section 19.21.3, Transmit Buffers).
15 to 11	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
10 to 0	CRC[10:0]	All 0	R/W	Header CRC (vRF!Header!HeaderCRC) Receive Buffer: Configuration not required Transmit Buffer: Header CRC calculated and configured by the Host For calculation of the header CRC the payload length of the frame send on the bus has to be considered. In static segment the payload length of all frames is configured by FRMHDC.SFDL[6:0].

19.12.4 FlexRay Write Header Section 3 (FRWRHS3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	DP[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 11	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
10 to 0	DP[10:0]	All 0	R/W	Data Pointer Pointer to the first 32-bit word of the data section of the addressed message buffer in the Message RAM.

19.12.5 FlexRay Input Buffer Command Mask (FRIBCM)

Configures how the message buffer in the Message RAM selected by register FRIBCR is updated. When IBF Host and IBF Shadow are swapped, also mask bits LHSH, LDSH, and STXRH are swapped with bits LHSS, LDSS, and STXRS to keep them attached to the respective Input Buffer transfer.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	STXRS	LDSS	LHSS
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	STXRH	LDSH	LHSH
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 19	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
18	STXRS	0	R	Set Transmission Request Shadow 0: Reset TXR flag 1: Set TXR flag, transmit buffer released for transmission (operation ongoing or finished)
17	LDSS	0	R	Load Data Section Shadow 0: Data section is not updated 1: Data section selected for transfer from Input Buffer to the Message RAM (transfer ongoing or finished)
16	LHSS	0	R	Load Header Section Shadow 0: Header section is not updated 1: Header section selected for transfer from Input Buffer to the Message RAM (transfer ongoing or finished)
15 to 3	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
2	STXRH	0	R/W	Set Transmission Request Host If this bit is set to '1', the TXR flag for the selected message buffer is set in the FRTXRQ1/2/3/4 registers to release the message buffer for transmission. In single-shot mode the flag is cleared by the CC after transmission has completed. TXR is evaluated for transmit buffers only. 0: Reset TXR flag 1: Set TXR flag, transmit buffer released for transmission
1	LDSH	0	R/W	Load Data Section Host 0: Data section is not updated 1: Data section selected for transfer from Input Buffer to the Message RAM
0	LHSH	0	R/W	Load Header Section Host 0: Header section is not updated 1: Header section selected for transfer from Input Buffer to the Message RAM

19.12.6 FlexRay Input Buffer Command Request (FRIBCR)

When the Host writes the number of the target message buffer in the Message RAM to IBRH[6:0], IBF Host and IBF Shadow are swapped. In addition the message buffer numbers stored under IBRH[6:0] and IBRS[6:0] are also swapped (see also section 19.24.2 (1), Data Transfer from Input Buffer to Message RAM).

With this write operation the IBSYS is set to '1'. The Message Handler then starts to transfer the contents of IBF Shadow to the message buffer in the Message RAM selected by IBRS[6:0].

While the Message Handler transfers the data from IBF Shadow to the target message buffer in the Message RAM, the Host may write the next message into the IBF Host. After the transfer between IBF Shadow and the Message RAM has completed, IBSYS is set back to '0' and the next transfer to the Message RAM may be started by the Host by writing the respective target message buffer number to IBRH[6:0].

If a write access to IBRH[6:0] occurs while IBSYS is '1', IBSYH is set to '1'. After completion of the ongoing data transfer from IBF Shadow to the Message RAM, IBF Host and IBF Shadow are swapped, IBSYH is reset to '0'. IBSYS remains set to '1', and the next transfer to the Message RAM is started. In addition the message buffer numbers stored under IBRH[6:0] and IBRS[6:0] are also swapped.

Any write access to an Input Buffer register while both IBSYS and IBSYH are set will cause the error flag FREIR.IIBA to be set. In this case the Input Buffer will not be changed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IB SYS	—	—	—	—	—	—	—	—	IBRS[6:0]						
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IB SYH	—	—	—	—	—	—	—	—	IBRH[6:0]						
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	IBSYS	0	R	Input Buffer Busy Shadow Set to '1' after writing IBRH[6:0]. When the transfer between IBF Shadow and the Message RAM has completed, IBSYS is set back to '0'. 0: Transfer between IBF Shadow and Message RAM completed 1: Transfer between IBF Shadow and Message RAM in progress
30 to 23	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
22 to 16	IBRS[6:0]	All 0	R	Input Buffer Request Shadow Number of the target message buffer actually updated/lately updated. Valid values are 0x00 to 0x7F (0...127).
15	IBSYH	0	R	Input Buffer Busy Host Set to '1' by writing IBRH[6:0] while IBSYS is still '1'. After the ongoing transfer between IBF Shadow and the Message RAM has completed, the IBSYH is set back to '0'. 0: No request pending 1: Request while transfer between IBF Shadow and Message RAM in progress
14 to 7	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
6 to 0	IBRH[6:0]	All 0	R/W	Input Buffer Request Host Selects the target message buffer in the Message RAM for data transfer from Input Buffer. Valid values are 0x00 to 0x7F (0...127).

19.13 Output Buffer (OBF)

Double buffer structure consisting of Output Buffer Host and Output Buffer Shadow. Used to read out message buffers from the Message RAM. While the Host can read from Output Buffer Host, the Message Handler transfers the selected message buffer from Message RAM to Output Buffer Shadow. The data transfer between Message RAM and Output Buffer (OBF) is described in section 19.24.2 (2), Data Transfer from Message RAM to Output Buffer.

19.13.1 FlexRay Read Data Section [1…64] (FRRDDSn)

Holds the data words read from the data section of the addressed message buffer. The data words (DWn) are read from the Message RAM in reception order from DW1 (byte0, byte1) to DWPL (PL = number of data words as defined by the payload length configured FRRDHS2.PLC[6:0]).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MD[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MD[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	MD[31:0]	All 0	R	Message Data (when byte swap is disabled: FXROC.FEC = 0) MD[7:0] = DW_{2n-1}, $byte_{4n-4}$ MD[15:8] = DW_{2n-1}, $byte_{4n-3}$ MD[23:16] = DW_{2n}, $byte_{4n-2}$ MD[31:24] = DW_{2n}, $byte_{4n-1}$ Message Data (when byte swap is enabled: FXROC.FEC = 1) MD[7:0] = DW_{2n}, $byte_{4n-1}$ MD[15:8] = DW_{2n}, $byte_{4n-2}$ MD[23:16] = DW_{2n-1}, $byte_{4n-3}$ MD[31:24] = DW_{2n-1}, $byte_{4n-4}$

Note: DW127 is located on FRRDDS64.MD[15:0]. In this case FRRDDS64.MD[31:16] is unused (no valid data). The Output Buffer RAMs are initialized to zero by a hardware or software reset or CHI command CLEAR_RAMS.

19.13.2 FlexRay Read Header Section 1 (FRRDHS1)

Values as configured via FRWRHS1

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	MBI	TXM	PPIT	CFG	CHB	CHA	—	CYC[6:0]						
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	FID[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31, 30	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
29	MBI	0	R	Message Buffer Interrupt The Message Buffer Interrupt bit set in FRWRHS1 can be read from this bit.
28	TXM	0	R	Transmission Mode The Transmission Mode bit set in FRWRHS1 can be read from this bit.
27	PPIT	0	R	Payload Preamble Indicator Transmit The Payload Preamble Indicator Transmit bit set in FRWRHS1 can be read from this bit.
26	CFG	0	R	Message Buffer Direction Configuration The Message Buffer Direction Configuration bit set in FRWRHS1 can be read from this bit.
25	CHB	0	R	Channel Filter Control The Channel Filter Control bit set in FRWRHS1 can be read from this bit.
24	CHA	0	R	
23	—	0	R	Reserved. This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
22 to 16	CYC[6:0]	All 0	R	Cycle Code The Cycle Code set in FRWRHS1 can be read from this bit.
15 to 11	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
10 to 0	FID[10:0]	All 0	R	Frame ID The Frame ID set in FRWRHS1 can be read from this bit.

In case that the message buffer read from the Message RAM belongs to the receive FIFO, FID[10:0] holds the received frame ID, while CYC[6:0], CHA, CHB, CFG, PPIT, TXM, and MBI are reset to '0'.

19.13.3 FlexRay Read Header Section 2 (FRRDHS2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	PLR[6:0]							—	PLC[6:0]						
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	CRC[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	—	0	R	Reserved. This bit is always read as 0. The write value should always be 0.
30 to 24	PLR[6:0]	All 0	R	Payload Length Received (vRF!Header!Length) Payload length value updated from received data frames (exception: if message buffer belongs to the receive FIFO PLR[6:0] is also updated from received null frames)
23	—	0	R	Reserved. This bit is always read as 0. The write value should always be 0.
22 to 16	PLC[6:0]	All 0	R	Payload Length Configured Length of data section (number of 2-byte words) as configured by the Host.
15 to 11	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
10 to 0	CRC[10:0]	All 0	R	Header CRC (vRF!Header!HeaderCRC) Receive Buffer: Header CRC updated from received data frames Transmit Buffer: Header CRC calculated and configured by the Host

When a message is stored into a message buffer the following behaviour with respect to payload length received and payload length configured is implemented:

PLR[6:0] > PLC[6:0]: The payload data stored in the message buffer is truncated to the payload length configured if PLC[6:0] even or else truncated to PLC[6:0] + 1.

PLR[6:0] ≤ PLC[6:0]: The received payload data is stored into the message buffers data section. The remaining data bytes of the data section as configured by PLC[6:0] are filled with undefined data

PLR[6:0] = zero: The message buffer's data section is filled with undefined data

PLC[6:0] = zero: Message buffer has no data section configured. No data is stored into the message buffer's data section.

Note: The Message RAM is organized in 4-byte words. When received data is stored into a message buffer's data section, the number of 2-byte data words written into the message buffer is PLC[6:0] rounded to the next even value. PLC[6:0] should be configured identical for all message buffers belonging to the receive FIFO. Header 2 is updated from data frames only.

19.13.4 FlexRay Read Header Section 3 (FRRDHS3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	RES	PPI	NFI	SYN	SFI	RCI	—	—	RCC[5:0]					
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	DP[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31, 30	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
29	RES	0	R	Reserved Bit Indicator (vRF!Header!Reserved) Reflects the state of the received reserved bit. The reserved bit is transmitted as '0'.
28	PPI	0	R	Payload Preamble Indicator (vRF!Header!PPIndicator) The payload preamble indicator defines whether a network management vector or message ID is contained within the payload segment of the received frame. 0: The payload segment of the received frame does not contain a network management vector nor a message ID 1: Static segment: Network management vector in the first part of the payload Dynamic segment: Message ID in the first part of the payload
27	NFI	0	R	Null Frame Indicator (vRF!Header!NFIndicator) Is set to '1' after storage of the first received data frame. 0: Up to now no data frame has been stored into the respective message buffer 1: At least one data frame has been stored into the respective message buffer

Bit	Bit Name	Initial Value	R/W	Description
26	SYN	0	R	Sync Frame Indicator (vRF!Header!SyFIndicator) A sync frame is marked by the sync frame indicator. 0: The received frame is not a sync frame 1: The received frame is a sync frame
25	SFI	0	R	Startup Frame Indicator (vRF!Header!SuFIndicator) A startup frame is marked by the startup frame indicator. 0: The received frame is not a startup frame 1: The received frame is a startup frame
24	RCI	0	R	Received on Channel Indicator (vSS!Channel) Indicates the channel from which the received data frame was taken to update the respective receive buffer. 0: Frame received on channel B 1: Frame received on channel A
23, 22	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
21 to 16	RCC[5:0]	All 0	R	Receive Cycle Count (vRF!Header!CycleCount) Cycle counter value updated from received data frame.
15 to 11	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
10 to 0	DP[10:0]	All 0	R	Data Pointer Pointer to the first 32-bit word of the data section of the addressed message buffer in the Message RAM.

Note: Header 3 is updated from data frames only.

19.13.5 FlexRay Message Buffer Status (FRMBS)

The message buffer status is updated by the CC with respect to the assigned channel(s) latest at the end of the slot following the slot assigned to the message buffer. The flags are updated only when the CC is in NORMAL_ACTIVE or NORMAL_PASSIVE state. If only one channel (A or B) is assigned to a message buffer, the channel-specific status flags of the other channel are written to zero. If both channels are assigned to a message buffer, the channel-specific status flags of both channels are updated. The message buffer status is updated only when the slot counter reached the configured frame ID and when the cycle counter filter matched. When the Host updates a message buffer via Input Buffer, all FRMBS flags are reset to zero independent of which FRIBCM bits are set or not. For details about receive/transmit filtering see sections 19.20, Filtering and Masking, 19.21, Transmit Process, and 19.22, Receive Process. Whenever the Message Handler changes one of the flags VFRA, VFRB, SEOA, SEOB, CEOA, CEOB, SVOA, SVOB, TCIA, TCIB, ESA, ESB, MLST, FTA, and FTB, the respective message buffer's MBC flag in registers FRMBSC1/2/3/4 is set.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	RESS	PPIS	NFIS	SYNS	SFIS	RCIS	—	—	CCS[5:0]					
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FTB	FTA	—	MLST	ESB	ESA	TCIB	TCIA	SVOB	SVOA	CEOB	CEOA	SEOB	SEOA	VFRB	VFRA
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31, 30	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
29	RESS	0	R	Reserved Bit Status (vRF!Header!Reserved) *1 Reflects the state of the received reserved bit. The reserved bit is transmitted as '0'.

Bit	Bit Name	Initial Value	R/W	Description
28	PPIS	0	R	Payload Preamble Indicator Status (vRF!Header!PPIndicator) *[1] The payload preamble indicator defines whether a network management vector or message ID is contained within the payload segment of the received frame. 0: The payload segment of the received frame does not contain a network management vector or a message ID 1: Static segment: Network management vector at the beginning of the payload Dynamic segment: Message ID at the beginning of the payload
27	NFIS	0	R	Null Frame Indicator Status (vRF!Header!NFIndicator) *[1] If set to '0' the payload segment of the received frame contains no usable data. 0: Received frame is a null frame 1: Received frame is not a null frame
26	SYNS	0	R	Sync Frame Indicator Status (vRF!Header!SyFIndicator) *[1] A sync frame is marked by the sync frame indicator. 0: No sync frame received 1: The received frame is a sync frame
25	SFIS	0	R	Startup Frame Indicator Status (vRF!Header!SuFIndicator) *[1] A startup frame is marked by the startup frame indicator. 0: No startup frame received 1: The received frame is a startup frame
24	RCIS	0	R	Received on Channel Indicator Status (vSS!Channel) *[1] Indicates the channel on which the frame was received. 0: Frame received on channel B 1: Frame received on channel A
23, 22	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
21 to 16	CCS[5:0]	All 0	R	Cycle Count Status Actual cycle count when status was updated.

Bit	Bit Name	Initial Value	R/W	Description
15	FTB	0	R	Frame Transmitted on Channel B *[2] Indicates that this node has transmitted a data frame in the configured slot on channel B. 0: No data frame transmitted on channel B 1: Data frame transmitted on channel B
14	FTA	0	R	Frame Transmitted on Channel A *[2] Indicates that this node has transmitted a data frame in the configured slot on channel A. 0: No data frame transmitted on channel A 1: Data frame transmitted on channel A
13	—	0	R	Reserved. This bit is always read as 0. The write value should always be 0.
12	MLST	0	R	Message Lost The flag is set in case the Host did not read the message before the message buffer was updated from a received data frame. Not affected by reception of null frames except for message buffers belonging to the receive FIFO. The flag is reset by a Host write to the message buffer via IBF or when a new message is stored into the message buffer after the message buffers ND flag was reset by reading out the message buffer via OBF. 0: No message lost 1: Unprocessed message was overwritten
11	ESB	0	R	Empty Slot Channel B In an empty slot there is no activity detected on the bus. The condition is checked in static and dynamic slots. 0: Bus activity detected in the assigned slot on channel B 1: No bus activity detected in the assigned slot on channel B

Bit	Bit Name	Initial Value	R/W	Description
10	ESA	0	R	Empty Slot Channel A In an empty slot there is no activity detected on the bus. The condition is checked in static and dynamic slots. 0: Bus activity detected in the assigned slot on channel A 1: No bus activity detected in the assigned slot on channel A
9	TCIB	0	R	Transmission Conflict Indication Channel B (vSS!TxConflictB) A transmission conflict indication is set if a transmission conflict has occurred on channel B. 0: No transmission conflict occurred on channel B 1: Transmission conflict occurred on channel B
8	TCIA	0	R	Transmission Conflict Indication Channel A (vSS!TxConflictA) A transmission conflict indication is set if a transmission conflict has occurred on channel A. 0: No transmission conflict occurred on channel A 1: Transmission conflict occurred on channel A
7	SVOB	0	R	Slot Boundary Violation Observed on Channel B (vSS!BViolationB) A slot boundary violation (channel active at the start or at the end of the assigned slot) was observed on channel B. 0: No slot boundary violation observed on channel B 1: Slot boundary violation observed on channel B
6	SVOA	0	R	Slot Boundary Violation Observed on Channel A (vSS!BViolationA) A slot boundary violation (channel active at the start or at the end of the assigned slot) was observed on channel A. 0: No slot boundary violation observed on channel A 1: Slot boundary violation observed on channel A

Bit	Bit Name	Initial Value	R/W	Description
5	CEOB	0	R	Content Error Observed on Channel B (vSS!ContentErrorB) A content error was observed in the assigned slot on channel B. 0: No content error observed on channel B 1: Content error observed on channel B
4	CEOA	0	R	Content Error Observed on Channel A (vSS!ContentErrorA) A content error was observed in the assigned slot on channel A. 0: No content error observed on channel A 1: Content error observed on channel A
3	SEOB	0	R	Syntax Error Observed on Channel B (vSS!SyntaxErrorB) A syntax error was observed in the assigned slot on channel B. 0: No syntax error observed on channel B 1: Syntax error observed on channel B
2	SEOA	0	R	Syntax Error Observed on Channel A (vSS!SyntaxErrorA) A syntax error was observed in the assigned slot on channel A. 0: No syntax error observed on channel A 1: Syntax error observed on channel A
1	VFRB	0	R	Valid Frame Received on Channel B (vSS!ValidFrameB) A valid frame indication is set if a valid frame was received on channel B. 0: No valid frame received on channel B 1: Valid frame received on channel B
0	VFRA	0	R	Valid Frame Received on Channel A (vSS!ValidFrameA) A valid frame indication is set if a valid frame was received on channel A. 0: No valid frame received on channel A 1: Valid frame received on channel A

Notes: 1. For receive buffers (CFG = '0') the following status bits are updated from both valid data and null frames. If no valid frame was received, the previous value is maintained. For transmit buffers the flags have no meaning and should be ignored.

2. The FlexRay protocol specification requires that FTA, and FTB can only be reset by the Host. Therefore the Cycle Count Status CCS[5:0] for these bits is only valid for the cycle where the bits are set to '1'.

19.13.6 FlexRay Output Buffer Command Mask (FROBCM)

Configures how the Output Buffer is updated from the message buffer in the Message RAM selected by register FROBCR. When OBF Host and OBF Shadow are swapped, also Mask bits RDSS and RHSS are copied to the register internal storage when a Message RAM transfer is requested by FROBCR.REQ. When OBF Host and OBF Shadow are swapped, mask bits RDSH and RHSH are swapped with the register internal storage to keep them attached to the respective Output Buffer transfer. The data transfer between Output Buffer and Message RAM is described in details in section 19.24.2(2), Data Transfer from Message RAM to Output Buffer.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RDSH	RHSH
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RDSS	RHSS
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
17	RDSH	0	R	Read Data Section Host 0: Data section is not read 1: Data section selected for transfer from Message RAM to Output Buffer
16	RHSH	0	R	Read Header Section Host 0: Header section is not read 1: Header section selected for transfer from Message RAM to Output Buffer
15 to 2	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
1	RDSS	0	R/W	Read Data Section Shadow 0: Data section is not read 1: Data section selected for transfer from Message RAM to Output Buffer

Bit	Bit Name	Initial Value	R/W	Description
0	RHSS	0	R/W	Read Header Section Shadow 0: Header section is not read 1: Header section selected for transfer from Message RAM to Output Buffer

Note: After the transfer of the header section from the Message RAM to OBF Shadow has completed, the message buffer status changed flag MBC of the selected message buffer in the FRMBSC1/2/3/4 registers is cleared. After the transfer of the data section from the Message RAM to OBF Shadow has completed, the new data flag ND of the selected message buffer in the FRNDAT1/2/3/4 registers is cleared.

19.13.7 FlexRay Output Buffer Command Request (FROBCR)

After setting bit REQ to '1' while OBSYS is '0', OBSYS is automatically set to '1', OBRS[6:0] is copied to the register internal storage, mask bits OBCM.RDSS and OBCM.RHSS are copied to register OBCM internal storage, and the transfer of the message buffer selected by OBRS[6:0] from the Message RAM to OBF Shadow is started. When the transfer between the Message RAM and OBF Shadow has completed, this is signalled by setting OBSYS back to '0'.

By setting bit VIEW to '1' while OBSYS is '0', OBF Host and OBF Shadow are swapped. Additionally mask bits FROBCM.RDSH and FROBCM.RHSH are swapped with the register OBCM internal storage to keep them attached to the respective Output Buffer transfer. OBRH[6:0] signals the number of the message buffer currently accessible by the Host.

If bits REQ and VIEW are set to '1' with the same write access while OBSYS is '0', OBSYS is automatically set to '1' and OBF Shadow and OBF Host are swapped. Additionally mask bits FROBCM.RDSH and FROBCM.RHSH are swapped with the registers internal storage to keep them attached to the respective Output Buffer transfer. Afterwards OBRS[6:0] is copied to the register internal storage, and the transfer of the selected message buffer from the Message RAM to OBF Shadow is started. While the transfer is ongoing the Host can read the message buffer transferred by the previous transfer from OBF Host. When the current transfer between Message RAM and OBF Shadow has completed, this is signalled by setting OBSYS back to '0'. Any write access to OBCR[15:8] while OBSYS is set will cause the error flag FREIR.IOBA to be set. In this case, the Output Buffer will not be changed. The data transfer between Output Buffer and Message RAM is described in details in section 19.24.2(2), Data Transfer from Message RAM to Output Buffer.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	OBRH[6:0]						
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OB SYS	—	—	—	—	—	REQ	VIEW	—	OBRS[6:0]						
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 23	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
22 to 16	OBRH[6:0]	All 0	R	Output Buffer Request Host Number of message buffer currently accessible by the Host via FRRDHS[1...3], FRMBS, and FRRDDS[1...64]. By writing VIEW to '1' OBF Shadow and OBF Host are swapped and the transferred message buffer is accessible by the Host. Valid values are 0x00 to 0x7F (0...127).
15	OBSYS	0	R	Output Buffer Busy Shadow Set to '1' after setting bit REQ. When the transfer between the Message RAM and OBF Shadow has completed, OBSYS is set back to '0'. 0: No transfer in progress 1: Transfer between Message RAM and OBF Shadow in progress
14 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	REQ	0	R/W	Request Message RAM Transfer Requests transfer of message buffer addressed by OBRS[6:0] from Message RAM to OBF Shadow. Only writeable while OBSYS = '0'. 0: No request 1: Transfer to OBF Shadow requested
8	VIEW	0	R/W	View Shadow Buffer Toggles between OBF Shadow and OBF Host. Only writeable while OBSYS = '0'. 0: No action 1: Swap OBF Shadow and OBF Host
7	—	0	R	Reserved. This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
6 to 0	OBRS[6:0]	All 0	R/W	Output Buffer Request Shadow Number of source message buffer to be transferred from the Message RAM to OBF Shadow. Valid values are 0x00 to 0x7F (0...127). If the number of the first message buffer of the receive FIFO is written to this register the Message Handler transfers the message buffer addressed by the GET Index (GIDX, see section 19.23, FIFO Function) to OBF Shadow.

19.14 Communication Cycle

A FlexRay communication cycle consists of the following elements:

- Static Segment
- Dynamic Segment (optional)
- Symbol Window (optional)
- Network Idle Time (NIT)

Static segment, dynamic segment, and symbol window form the Network Communication Time (NCT). For each communication channel the slot counter starts at 1 and counts up until the end of the dynamic segment is reached. Both channels share the same arbitration grid which means that they use the same synchronized macrotick.

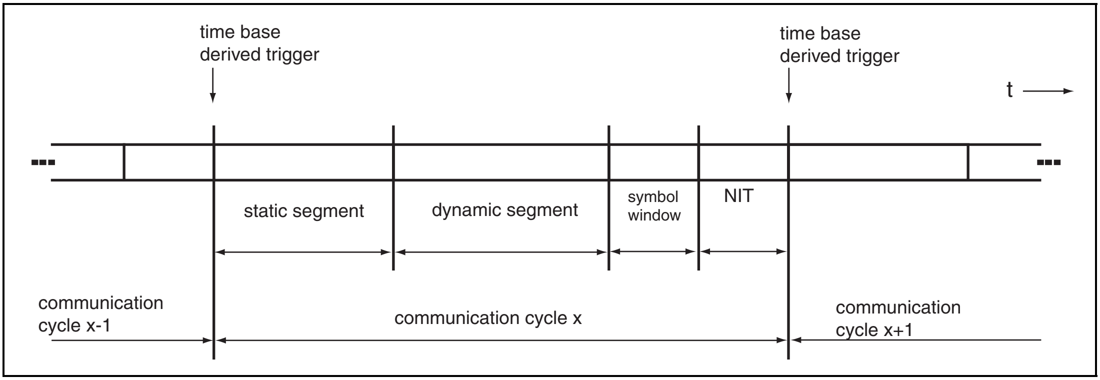


Figure 19.2 Structure of Communication Cycle

Communication on FlexRay networks is based on frames and symbols. The wakeup symbol (WUS) and the collision avoidance symbol (CAS) are transmitted outside the communication cycle to setup the time schedule. Frames and media access test symbols (MTS) are transmitted inside the communication cycle.

19.14.1 Static Segment

The Static Segment is characterized by the following features:

- Time slots of fixed length (optionally protected by bus guardian)
- Start of frame transmission at action point of the respective static slot
- Payload length same for all frames on both channels

Parameters: Number of Static Slots FRGTUC7.NSS[9:0], Static Slot Length FRGTUC7.SSL[9:0], Payload Length Static FRMHDC.SFDL[6:0], Action Point Offset FRGTUC9.APO[5:0]

19.14.2 Dynamic Segment

The Dynamic Segment is characterized by the following features:

- All controllers have bus access (no bus guardian protection possible)
- Variable payload length and duration of slots, different for both channels
- Start of transmission at minislot action point

Parameters: Number of Minislots FRGTUC8.NMS[12:0], Minislot Length FRGTUC8.MSL[5:0], Minislot Action Point Offset FRGTUC9.MAPO[4:0], Start of Latest Transmit (last minislot) FRMHDC.SLT[12:0]

19.14.3 Symbol Window

During the symbol window only one media access test symbol (MTS) may be transmitted per channel. MTS symbols are sent in NORMAL_ACTIVE state to test the bus guardian.

The symbol window is characterized by the following features:

- Send single symbol
- Transmission of the MTS symbol starts at the symbol windows action point

Parameters: Symbol Window Action Point Offset FRGTUC9.APO[5:0] (same as for static slots), Network Idle Time Start FRGTUC4.NIT[13:0]

19.14.4 Network Idle Time (NIT)

During network idle time the CC has to perform the following tasks:

- Calculate clock correction terms (offset and rate)
- Distribute offset correction over multiple macroticks after offset correction start
- Perform cluster cycle related tasks

Parameters: Network Idle Time Start FRGTUC4.NIT[13:0], Offset Correction Start FRGTUC4.OCS[13:0]

19.14.5 Configuration of NIT Start and Offset Correction Start

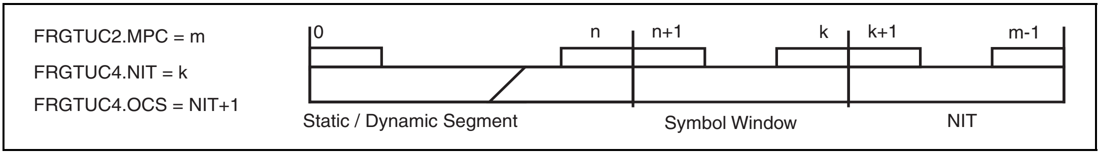


Figure 19.3 Configuration of NIT Start and Offset Correction Start

The number of macroticks per cycle gMacroPerCycle is assumed to be m. It is configured by programming FRGTUC2.MPC = m.

The static/dynamic segment starts with macrotick 0 and ends with macrotick n:

n = static segment length + dynamic segment offset + dynamic segment length - 1MT
n = gNumberOfStaticSlots × gdStaticSlot + dynamic segment offset
+ gNumberOfMinislots × gdMinislot - 1 MT

The static segment length is configured by FRGTUC7.SSL[9:0] and FRGTUC7.NSS[9:0].

The dynamic segment length is configured by FRGTUC8.MSL[5:0] and FRGTUC8.NMS[12:0].

The dynamic segment offset is:
If gdActionPointOffset ≤ gdMinislotActionPointOffset:
dynamic segment offset = 0 MT
Else if gdActionPointOffset > gdMinislotActionPointOffset:
dynamic segment offset = gdActionPointOffset - gdMinislotActionPointOffset

The NIT starts with macrotick k+1 and ends with the last macrotick of cycle m-1. It has to be configured by setting FRGTUC4.NIT[13:0] = k.

For the FlexRay, the offset correction start is required to be FRGTUC4.OCS[13:0] ≥ FRGTUC4.NIT[13:0] + 1 = k + 1.

The length of symbol window results from the number of macroticks between the end of the static/dynamic segment and the beginning of the NIT. It can be calculated by k - n.

19.15 Communication Modes

The FlexRay Protocol Specification v2.1 defines the Time-Triggered Distributed (TT-D) mode.

19.15.1 Time-Triggered Distributed (TT-D)

In TT-D mode the following configurations are possible:

- Pure static: Minimum 2 static slots + symbol window (optional)
- Mixed static/dynamic: Minimum 2 static slots + dynamic segment + symbol window (optional)

A minimum of two coldstart nodes needs to be configured for distributed time-triggered operation.

Two fault-free coldstart nodes are necessary for the cluster startup. Each startup frame must be a sync frame, therefore all coldstart nodes are sync nodes.

19.16 Clock Synchronization

In TT-D mode a distributed clock synchronization is used. Each node individually synchronizes itself to the cluster by observing the timing of received sync frames from other nodes.

19.16.1 Global Time

Activities in a FlexRay node, including communication, are based on the concept of a global time, even though each individual node maintains its own view of it. It is the clock synchronization mechanism that differentiates the FlexRay cluster from other node collections with independent clock mechanisms. The global time is a vector of two values; the cycle (cycle counter) and the cycle time (macrotick counter).

Cluster specific:

- Macrotick (MT) = basic unit of time measurement in a FlexRay network, a macrotick consists of an integer number of microticks (µT)
- Cycle length = duration of a communication cycle in units of macroticks (MT)

19.16.2 Local Time

Internally, nodes time their behaviour with microtick resolution. Microticks are time units derived from the oscillator clock tick of the specific node. Therefore microticks are controller-specific units. They may have different duration in different controllers. The precision of a node's local time difference measurements is a microtick (μT).

Node specific:

- Oscillator clock -> prescaler -> microtick (μT)
- μT = basic unit of time measurement in a CC, clock correction is done in units of μTs
- Cycle counter + macrotick counter = nodes local view of the global time

19.16.3 Synchronization Process

Clock synchronization is performed by means of sync frames. Only preconfigured nodes (sync nodes) are allowed to send sync frames. In a two-channel cluster a sync node has to send its sync frame on both channels.

For synchronization in FlexRay the following constraints have to be considered:

- Max. one sync frame per node in one communication cycle
- Max. 15 sync frames per cluster in one communication cycle
- Every node has to use a preconfigured number of sync frames (FRGTUC2.SNM[3:0]) for clock synchronization
- Minimum of two sync nodes required for clock synchronization and startup

For clock synchronization the time difference between expected and observed arrival time of sync frames received during the static segment is measured. In a two channel cluster the sync node has to be configured to send sync frames on both channels. The calculation of correction terms is done during NIT (offset: every cycle, rate: every odd cycle) by using an FTM algorithm. For details see FlexRay protocol specification v2.1, chapter 8.

(1) Offset (Phase) Correction

- Only deviation values measured and stored in the current cycle used
- For a two channel node the smaller value will be taken
- Calculation during NIT of every communication cycle
- Offset correction value calculated in even cycles used for error checking only
- Checked against limit values

- Correction value is a signed integer number of µTs
- Correction done in odd numbered cycles, distributed over the macroticks beginning at offset correction start up to cycle end (end of NIT) to shift nodes next start of cycle (MTs lengthened/shortened)

(2) Rate (Frequency) Correction

- Pairs of deviation values measured and stored in even/odd cycle pair used
- For a two channel node the average of the differences from the two channels is used
- Calculated during NIT of odd numbered cycles
- Cluster drift damping is performed using global damping value
- Checked against limit values
- Correction value is a signed integer number of µTs
- Distributed over macroticks comprising the next even/odd cycle pair (MTs lengthened/shortened)

(3) Sync Frame Transmission

Sync frame transmission is only possible from buffer 0 and 1. Message buffer 1 may be used for sync frame transmission in case that sync frames should have different payloads on the two channels. In this case, bit FRMRC.SPLM has to be programmed to '1'.

Message buffers used for sync frame transmission have to be configured with the key slot ID and can be (re)configured in DEFAULT_CONFIG or CONFIG state only. For nodes transmitting sync frames FRSUCC1.TXSY must be set to '1'.

19.16.4 External Clock Synchronization

During normal operation, independent clusters can drift significantly. If synchronous operation across independent clusters is desired, external synchronization is necessary; even though the nodes within each cluster are synchronized. This can be accomplished with synchronous application of host-deduced rate and offset correction terms to the clusters.

- External offset/rate correction value is a signed integer
- External offset/rate correction value is added to calculated offset/rate correction value
- Aggregated offset/rate correction term (external + internal) is not checked against configured limits

19.17 Error Handling

The implemented error handling concept is intended to ensure that, in case of a lower layer protocol error in one single node, communication between non-affected nodes can be maintained. In some cases, higher layer program activity is required for the CC to resume normal operation. A change of the error handling state will set FREIR.PEMC and may trigger an interrupt to the Host if enabled. The actual error mode is signalled by FRCCEV.ERRM[1:0].

Table 19.5 POC Error Modes (Degradation Model)

Error Mode	Activity
ACTIVE (green)	Full operation State: NORMAL_ACTIVE The CC is fully synchronized and supports the cluster wide clock synchronization. The host is informed of any error condition(s) or status change by interrupt (if enabled) or by reading the error and status interrupt flags from registers FREIR and FRSIR.
PASSIVE (yellow)	Reduced operation State: NORMAL_PASSIVE, CC self rescue allowed The CC stops transmitting frames and symbols, but received frames are still processed. Clock synchronization mechanisms are continued based on received frames. No active contribution to the cluster wide clock synchronization. The host is informed of any error condition(s) or status change by interrupt (if enabled) or by reading the error and status interrupt flags from registers FREIR and FRSIR.
COMM_HALT (red)	Operation halted State: HALT, CC self rescue not allowed The CC stops frame and symbol processing, clock synchronization processing, and the macrotick generation. The host has still access to error and status information by reading the error and status interrupt flags from registers FREIR and FRSIR. The bus drivers are disabled.

19.17.1 Clock Correction Failed Counter

When the clock correction failed counter reaches the "maximum without clock correction passive" limit defined by FRSUCC3.WCP[3:0], the POC transits from NORMAL_ACTIVE to NORMAL_PASSIVE state. When it reaches the "maximum without clock correction fatal" limit defined by FRSUCC3.WCF[3:0], it transits from NORMAL_ACTIVE or NORMAL_PASSIVE to HALT state.

The clock correction failed counter FRCCEV.CCFC[3:0] allows the Host to monitor the duration of the inability of a node to compute clock correction terms after the CC passed protocol startup phase. It will be incremented by one at the end of any odd communication cycle during which either the missing offset correction FRSFS.MOCS or the missing rate correction FRSFS.MRCS flag is set.

The clock correction failed counter is reset to zero at the end of an odd communication cycle if neither the missing offset correction FRSFS.MOCS nor the missing rate correction FRSFS.MRCS flag is set.

The clock correction failed counter stops incrementing when the "maximum without clock correction fatal" value FRSUCC3.WCF[3:0] is reached (i.e. incrementing the counter at its maximum value will not cause it to wrap around back to zero). The clock correction failed counter is initialized to zero when the CC enters READY state or when NORMAL_ACTIVE state is entered.

Note: The transition to HALT state is prevented if FRSUCC1.HCSE is not set.

19.17.2 Passive to Active Counter

The passive to active counter controls the transition of the POC from NORMAL_PASSIVE to NORMAL_ACTIVE state. FRSUCC1.PTA[4:0] defines the number of consecutive even/odd cycle pairs that must have valid clock correction terms before the CC is allowed to transit from NORMAL_PASSIVE to NORMAL_ACTIVE state. If FRSUCC1.PTA[4:0] is set to zero the CC is not allowed to transit from NORMAL_PASSIVE to NORMAL_ACTIVE state.

19.17.3 HALT Command

In case the Host wants to stop FlexRay communication of the local node it can bring the CC into HALT state by asserting the HALT command. This can be done by writing FRSUCC1.CMD[3:0] = "0110". In order to shut down communication on an entire FlexRay network, a higher layer protocol is required to assure that all nodes apply the HALT command at the same time.

The POC state from which the transition to HALT state took place can be read from FRCCSV.PSL[5:0].

When called in NORMAL_ACTIVE or NORMAL_PASSIVE state the POC transits to HALT state at the end of the current cycle. When called in any other state FRSUCC1.CMD[3:0] will be reset to "0000" = command_not_accepted and bit FREIR.CNA is set to '1'. If enabled an interrupt to the Host is generated.

19.17.4 FREEZE Command

In case the Host detects a severe error condition it can bring the CC into HALT state by asserting the FREEZE command. This can be done by writing FRSUCC1.CMD[3:0] = "0111". The FREEZE command triggers the entry of the HALT state immediately regardless of the actual POC state.

The POC state from which the transition to HALT state took place can be read from FRCCSV.PSL[5:0].

19.18 Communication Controller States

19.18.1 Communication Controller State Diagram

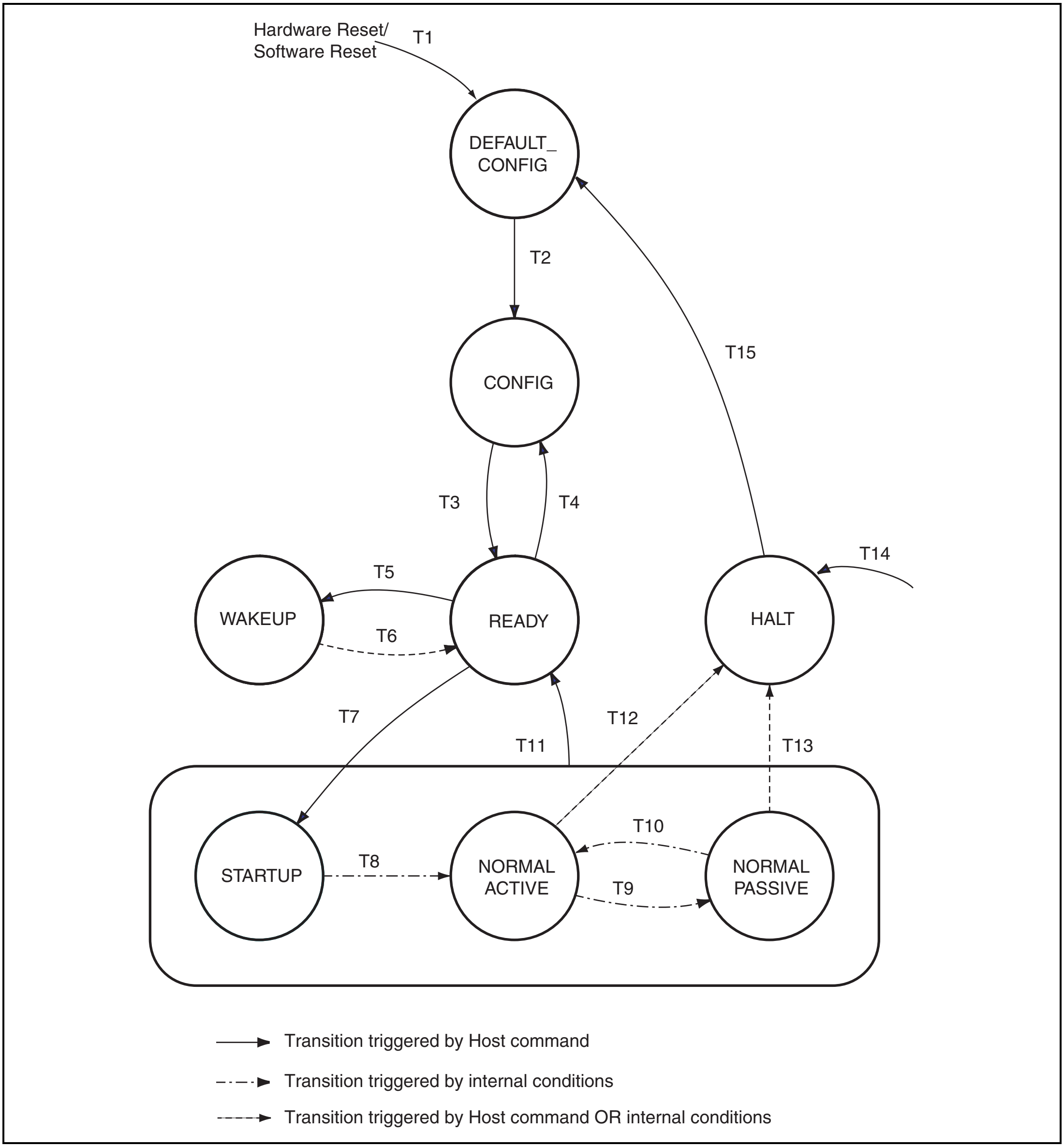

Hardware Reset/
Software Reset
T1
DEFAULT_
CONFIG
T2
CONFIG
T15
T3
T4
T14
T5
WAKEUP
READY
HALT
T6
T7
T11
T12
T13
T10
STARTUP
T8
NORMAL
ACTIVE
T9
NORMAL
PASSIVE
Transition triggered by Host command
Transition triggered by internal conditions
Transition triggered by Host command OR internal conditions

Figure 19.4 Overall State Diagram of FlexRay Communication Controller

State transitions are controlled by the $\overline{\text{RES}}$ pin, FRRxD_A pin, FRRxD_B pin, POC state machine, CHI Command Vector (FRSUCC1.CMD[3:0]), and also by FlexRay enable bit (FOPEN) in the FlexRay operating control register.

The CC exits from all states to HALT state after execution of the FREEZE command (FRSUCC1.CMD[3:0] = "0111").

Table 19.6 State Transitions of FlexRay Overall State Machine

T#	Condition	From	To
1	Reset	All states	DEFAULT_CONFIG
2	CONFIG command issued (FRSUCC1.CMD[3:0] = "0001")	DEFAULT_CONFIG	CONFIG
3	Unlock sequence followed by READY command (FRSUCC1.CMD[3:0] = "0010")	CONFIG	READY
4	CONFIG command issued (FRSUCC1.CMD[3:0] = "0001")	READY	CONFIG
5	WAKEUP command issued (FRSUCC1.CMD[3:0] = "0011")	READY	WAKEUP
6	Wakeup pattern transmission completed, wakeup pattern reception completed, frame header reception completed, wakeup collision detected, or READY command issued (FRSUCC1.CMD[3:0] = "0010")	WAKEUP	READY
7	RUN command issued (FRSUCC1.CMD[3:0] = "0100")	READY	STARTUP
8	Successful startup	STARTUP	NORMAL_ACTIVE
9	Clock correction failed counter reached the value set in FRSUCC3.WCP[3:0]	NORMAL_ACTIVE	NORMAL_PASSIVE
10	Number of valid clock correction cycle pairs reached the value set in FRSUCC1.PTA[4:0]	NORMAL_PASSIVE	NORMAL_ACTIVE
11	READY command issued (FRSUCC1.CMD[3:0] = "0010")	STARTUP, NORMAL_ACTIVE, NORMAL_PASSIVE	READY
12	Clock correction failed counter reached the value set in FRSUCC3.WCF[3:0] when bit FRSUCC1.HCSE is set to '1' or HALT command issued (FRSUCC1.CMD[3:0] = "0110")	NORMAL_ACTIVE	HALT
13	Clock correction failed counter reached the value set in FRSUCC3.WCF[3:0] when bit FRSUCC1.HCSE is set to '1' or HALT command issued (FRSUCC1.CMD[3:0] = "0110")	NORMAL_PASSIVE	HALT
14	FREEZE command issued (FRSUCC1.CMD[3:0] = "0111")	All states	HALT
15	CONFIG command issued (FRSUCC1.CMD[3:0] = "0001")	HALT	DEFAULT_CONFIG

19.18.2 DEFAULT_CONFIG State

In DEFAULT_CONFIG state, the CC is stopped. All configuration registers are accessible and the pins to the physical layer are in their inactive state.

The CC enters this state

- When leaving reset
- When exiting from HALT state

To leave DEFAULT_CONFIG state the Host has to write FRSUCC1.CMD[3:0] = "0001". The CC then transits to CONFIG state.

19.18.3 CONFIG State

In CONFIG state, the CC is stopped. All configuration registers are accessible and the pins to the physical layer are in their inactive state. This state is used to initialize the CC configuration.

The CC enters this state

- When exiting from DEFAULT_CONFIG state
- When exiting from READY state

When the state has been entered via HALT and DEFAULT_CONFIG state, the Host can analyze status information and configuration. Before leaving CONFIG state the Host has to assure that the configuration is fault-free.

To leave CONFIG state, the Host has to perform the unlock sequence as described in section 19.6.5, FlexRay Lock Register (FRLCK). Directly after unlocking the CONFIG state the Host has to write FRSUCC1.CMD[3:0] to enter the next state.

Note: Status bits FRMHDS[14:0], registers FRTXRQ1/2/3/4, and status data stored in the Message RAM are not affected by the transition of the POC from CONFIG to READY state.

When the CC is in CONFIG state it is also possible to bring the CC into a power saving mode by halting the module clocks (sampling clock or peripheral bus clock). To do this the Host has to assure that all Message RAM transfers have finished before turning off the clocks.

19.18.4 READY State

After unlocking CONFIG state and writing FRSUCC1.CMD[3:0] = "0010" the CC enters READY state. From this state the CC can transit to WAKEUP state and perform a cluster wakeup or to STARTUP state to perform a coldstart or to integrate into a running cluster.

The CC enters this state

- When exiting from CONFIG, WAKEUP, STARTUP, NORMAL_ACTIVE, or NORMAL_PASSIVE state by writing FRSUCC1.CMD[3:0] = "0010" (READY command).

The CC exits from this state

- To CONFIG state by writing FRSUCC1.CMD[3:0] = "0001" (CONFIG command)
- To WAKEUP state by writing FRSUCC1.CMD[3:0] = "0011" (WAKEUP command)
- To STARTUP state by writing FRSUCC1.CMD[3:0] = "0100" (RUN command)

Internal counters and the CC status flags are reset when the CC enters STARTUP state.

Note: Status bits FRMHDS[14:0], registers FRTXRQ1/2/3/4, and status data stored in the Message RAM are not affected by the transition of the POC from READY to STARTUP state.

19.18.5 WAKEUP State

The description below is intended to help configuring wakeup for the FlexRay IP-module. A detailed description of the wakeup procedure together with the respective SDL diagrams can be found in the FlexRay protocol specification v2.1, section 7.1.

The CC enters this state

- When exiting from READY state by writing FRSUCC1.CMD[3:0] = "0011" (WAKEUP command).

The CC exits from this state to READY state

- After complete non-aborted transmission of wakeup pattern
- After WUP reception
- After detecting a WUP collision
- After reception of a frame header
- By writing FRSUCC1.CMD[3:0] = "0010" (READY command)

The cluster wakeup must precede the communication startup in order to ensure that all nodes in a cluster are awake. The minimum requirement for a cluster wakeup is that all bus drivers are supplied with power. A bus driver has the ability to wake up the other components of its node when it receives a wakeup pattern on its channel. At least one node in the cluster needs an external wakeup source.

The Host completely controls the wakeup procedure. It is informed about the state of the cluster by the bus driver and the CC and configures bus guardian (if available) and CC to perform the cluster wakeup. The CC provides to the Host the ability to transmit a special wakeup pattern on each of its available channels separately. The CC needs to recognize the wakeup pattern only during WAKEUP state.

Wakeup may be performed on only one channel at a time. The Host has to configure the wakeup channel while the CC is in CONFIG state by writing FRSUCC1.WUCS. The CC ensures that ongoing communication on this channel is not disturbed. The CC cannot guarantee that all nodes connected to the configured channel awake upon the transmission of the wakeup pattern, since these nodes cannot give feedback until the startup phase. The wakeup procedure enables single-channel devices in a two-channel system to trigger the wakeup, by only transmitting the wakeup pattern on the single channel to which they are connected. Any coldstart node that deems a system startup necessary will then wake the remaining channel before initiating communication startup.

The wakeup procedure tolerates any number of nodes simultaneously trying to wakeup a single channel and resolves this situation such that only one node transmits the pattern. Additionally the wakeup pattern is collision resilient, so even in the presence of a fault causing two nodes to simultaneously transmit a wakeup pattern, the resulting collided signal can still wake the other nodes.

After wakeup the CC returns to READY state and signals the change of the wakeup status to the Host by setting flag FRSIR.WST. The wakeup status vector can be read from FRCCSV.WSV[2:0]. If a valid wakeup pattern was received also either flag FRSIR.WUPA or flag FRSIR.WUPB is set.

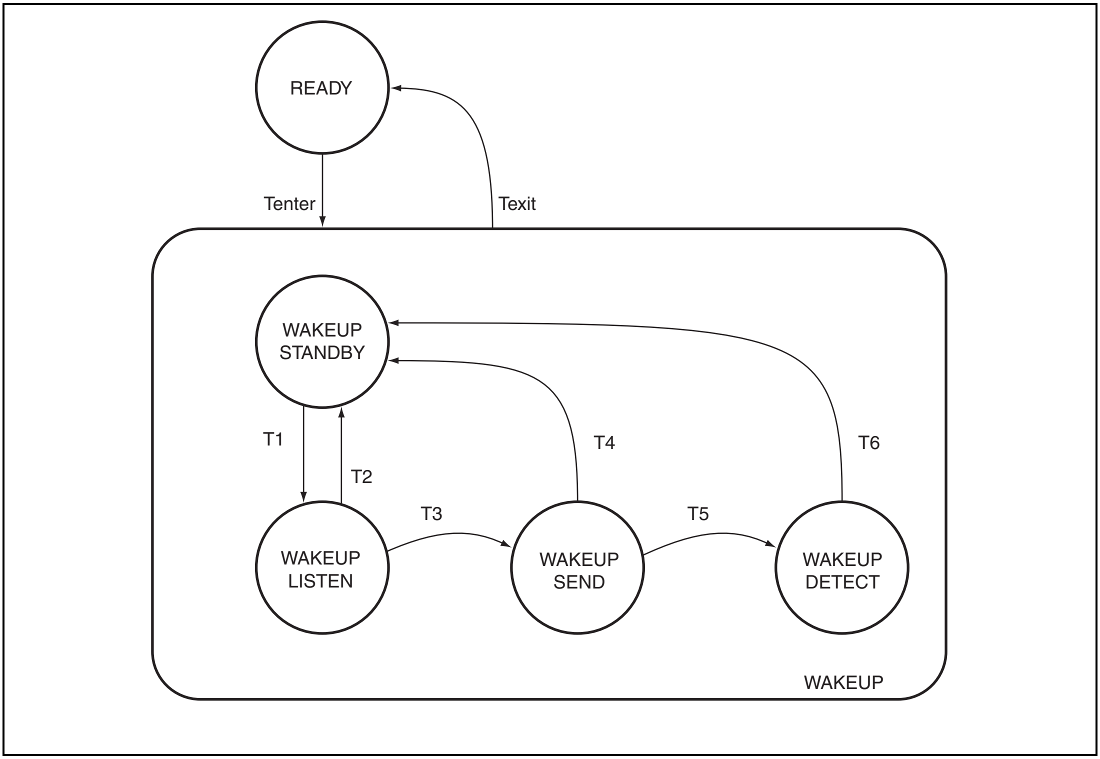


Figure 19.5 State Transitions Regarding WAKEUP State

Table 19.7 State Transitions Regarding WAKEUP State

T#	Condition	From	To
enter	Host commands change to WAKEUP state by writing FRSUCC1.CMD[3:0] = "0011" (WAKEUP command)	READY	WAKEUP
1	CHI command WAKEUP triggers wakeup FSM to transit to WAKEUP_LISTEN state	WAKEUP_STANDBY	WAKEUP_LISTEN
2	Received WUP on wakeup channel selected by bit FRSUCC1.WUCS OR frame header on either available channel	WAKEUP_LISTEN	WAKEUP_STANDBY
3	Timer event	WAKEUP_LISTEN	WAKEUP_SEND
4	Complete, non-aborted transmission of wakeup pattern	WAKEUP_SEND	WAKEUP_STANDBY
5	Collision detected	WAKEUP_SEND	WAKEUP_DETECT
6	Wakeup timer expired OR WUP detected on wakeup channel selected by bit FRSUCC1.WUCS OR frame header received on either available channel	WAKEUP_DETECT	WAKEUP_STANDBY
exit	Wakeup completed (after T2 or T4 or T6) OR Host commands change to READY state by writing FRSUCC1.CMD[3:0] = "0010" (READY command). This command also resets the wakeup FSM to WAKEUP_STANDBY state	WAKEUP	READY

The WAKEUP_LISTEN state is controlled by the wakeup timer and the wakeup noise timer. The two timers are controlled by the parameters listen timeout FRSUCC2.LT[20:0] and listen timeout noise FRSUCC2.LTN[3:0]. Listen timeout enables a fast cluster wakeup in case of a noise free environment, while listen timeout noise enables wakeup under more difficult conditions regarding noise interference.

In WAKEUP_SEND state the CC transmits the wakeup pattern on the configured channel and checks for collisions. After return from wakeup the Host has to bring the CC into STARTUP state by CHI command RUN.

In WAKEUP_DETECT state the CC attempts to identify the reason for the wakeup collision detected in WAKEUP_SEND state. The monitoring is bounded by the expiration of listen timeout as configured by FRSUCC2.LT[20:0]. Either the detection of a wakeup pattern indicating a wakeup attempt by another node or the reception of a frame header indicating ongoing communication, causes the direct transition to READY state. Otherwise WAKEUP_DETECT is left after expiration of listen timeout; in this case the reason for wakeup collision is unknown.

The Host has to be aware of possible failures of the wakeup and act accordingly. It is advisable to delay any potential startup attempt of the node having instigated the wakeup by the minimal time it takes another coldstart node to become awake and to be configured.

The FlexRay Protocol Specification v2.1 recommends that two different CCs shall awake the two channels.

(1) Host Activities

The host must coordinate the wakeup of the two channels and must decide whether, or not, to wake a specific channel. The sending of the wakeup pattern is initiated by the Host. The wakeup pattern is detected by the remote BDs and signalled to their local Host.

Wakeup procedure controlled by Host (single-channel wakeup):

- Configure the CC in CONFIG state
  - Select wakeup channel by programming bit FRSUCC1.WUCS
- Check local BDs whether a WUP was received
- Activate BD of selected wakeup channel
- Command CC to enter READY state
- Command CC to start wakeup on the configured channel by writing FRSUCC1.CMD[3:0] = "0011"
  - CC enters WAKEUP
  - CC returns to READY state and signals status of wakeup attempt to the Host
- Wait predefined time to allow the other nodes to wakeup and configure themselves
- Coldstart node:
  - In a dual channel cluster wait for WUP on the other channel
  - Reset coldstart inhibit flag FRCCSV.CSI by writing FRSUCC1.CMD[3:0] = "1001" (ALLOW_COLDSTART command)
- Command CC to enter startup by writing FRSUCC1.CMD[3:0] = "0100" (RUN command)

Wakeup procedure triggered by BD:

- Wakeup recognized by BD
- BD triggers power-up of Host (if required)
- BD signals wakeup event to Host
- Host configures its local CC
- If necessary, Host commands wakeup of second channel and waits predefined time to allow the other nodes to wakeup and configure themselves
- Host commands CC to enter STARTUP state by writing FRSUCC1.CMD[3:0] = "0100" (RUN command)

(2) Wakeup Pattern (WUP)

The wakeup pattern (WUP) is composed of at least two wakeup symbols (WUS). Wakeup symbol and wakeup pattern are configured by registers FRPRTC1 and FRPRTC2.

- Single channel wakeup, wakeup symbol may not be sent on both channels at the same time
- Wakeup symbol collision resilient for at least two sending nodes (two overlapping wakeup symbols always recognizable)
- Wakeup symbol must be configured identical in all nodes of a cluster
- Wakeup symbol transmit low time configured by FRPRTC2.TXL[5:0]
- Wakeup symbol idle time used to listen for activity on the bus, configured by FRPRTC2.TXI[7:0]
- A wakeup pattern composed of at least two Tx-wakeup symbols needed for wakeup
- Number of repetitions configurable by FRPRTC1.RWP[5:0] (2 to 63 repetitions)
- Wakeup symbol receive window length configured by FRPRTC1.RXW[8:0]
- Wakeup symbol receive low time configured by FRPRTC2.RXL[5:0]
- Wakeup symbol receive idle time configured by FRPRTC2.RXI[5:0]

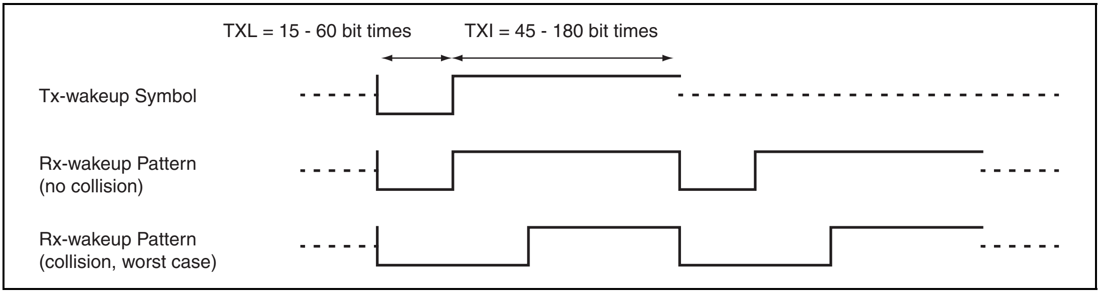


Figure 19.6 Timing of Wakeup Pattern

19.18.6 STARTUP State

The description below is intended to help configuring startup for the FlexRay IP-module. A detailed description of the startup procedure together with the respective SDL diagrams can be found in the FlexRay protocol specification v2.1, section 7.2.

Any node entering STARTUP state that has coldstart capability should assure that both channels attached have been awakened before initiating coldstart.

It cannot be assumed that all nodes and stars need the same amount of time to become completely awake and to be configured. Since at least two nodes are necessary to start up the cluster communication, it is advisable to delay any potential startup attempt of the node having instigated the wakeup by the minimal amount of time it takes another coldstart node to become awake, to be configured and to enter startup. It may require several hundred milliseconds (depending on the hardware used) before all nodes and stars are completely awakened and configured.

Startup is performed on all channels synchronously. During startup, a node only transmits startup frames. Startup frames are both sync frames and null frames during startup.

A fault-tolerant, distributed startup strategy is specified for initial synchronization of all nodes. In general, a node may enter NORMAL_ACTIVE state via (see figure 19.7):

- Coldstart path initiating the schedule synchronization (leading coldstart node)
- Coldstart path joining other coldstart nodes (following coldstart node)
- Integration path integrating into an existing communication schedule (all other nodes)

A coldstart attempt begins with the transmission of a collision avoidance symbol (CAS). Only a cold-start node that had transmitted the CAS transmits frames in the first four cycles after the CAS, it is then joined firstly by the other coldstart nodes and afterwards by all other nodes.

A coldstart node has bits FRSUCC1.TXST and FRSUCC1.TXSY set to '1'. Message buffer 0 holds the key slot ID which defines the slot number where the startup frame is send. In the frame header of the startup frame the startup frame indicator bit is set.

In clusters consisting of three or more nodes, at least three nodes shall be configured to be coldstart nodes. In clusters consisting of two nodes, both nodes must be coldstart nodes. At least two fault-free coldstart nodes are necessary for the cluster to startup.

Each startup frame must also be a sync frame; therefore each coldstart node will also be a sync node. The number of coldstart attempts is configured by FRSUCC1.CSA[4:0].

A non-coldstart node requires at least two startup frames from distinct nodes for integration. It may start integration before the coldstart nodes have finished their startup. It will not finish its startup until at least two coldstart nodes have finished their startup.

Both non-coldstart nodes and coldstart nodes start passive integration via the integration path as soon as they receive sync frames from which to derive the TDMA schedule information. During integration, the node has to adapt its own clock to the global clock (rate and offset) and has to make its cycle time consistent with the global schedule observable at the network. Afterwards, these settings are checked for consistency with all available network nodes. The node can only leave the integration phase and actively participate in communication when these checks are passed.

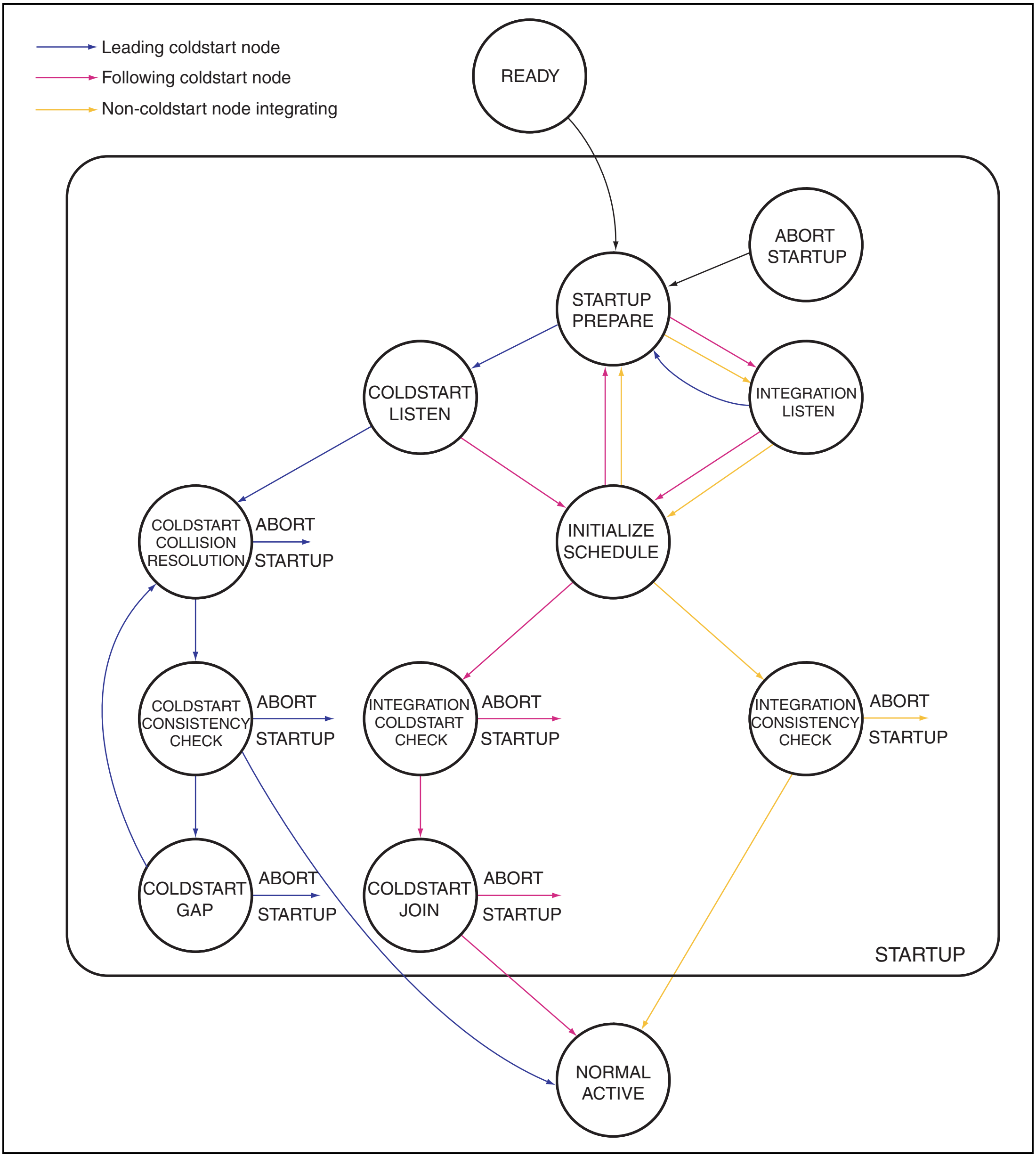


Figure 19.7 State Diagram Time-Triggered Startup

(1) Coldstart Inhibit Mode

In coldstart inhibit mode the node is prevented from initializing the TDMA communication schedule. If bit FRCCSV.CSI is set, the node is not allowed to initialize the cluster communication, i.e. entering the coldstart path is prohibited. The node is allowed to integrate to a running cluster or to transmit startup frames after another coldstart node started the initialization of the cluster communication.

The coldstart inhibit bit FRCCSV.CSI is set whenever the POC enters READY state. The bit has to be cleared under control of the Host by CHI command ALLOW_COLDSTART (FRSUCC1.CMD[3:0] = "1001")

(2) Startup Timeouts

The CC supplies two different µT timers supporting two timeout values, startup timeout and startup noise timeout. The two timers are started when the CC enters the COLDSTART_LISTEN state. The expiration of either of these timers causes the node to leave the initial sensing phase (COLDSTART_LISTEN state) with the intention of starting up communication.

Note: The startup and startup noise timers are identical with the wakeup and wakeup noise timers and use the same configuration values FRSUCC2.LT[20:0] and FRSUCC2.LTN[3:0].

Startup Timeout

The startup timeout limits the listen time used by a node to determine if there is already communication between other nodes or at least one coldstart node actively requesting the integration of others. The startup timer is configured by programming FRSUCC2.LT[20:0] (see section 19.8.2, FlexRay SUC Configuration Register 2 (FRSUCC2)).

The startup timeout is: pdListenTimeout = FRSUCC2.LT[20:0]

The startup timer is restarted upon:

- Entering the COLDSTART_LISTEN state
- Both channels reaching idle state while in COLDSTART_LISTEN state

The startup timer is stopped:

- If communication channel activity is detected on one of the configured channels while the node is in the COLDSTART_LISTEN state
- When the COLDSTART_LISTEN state is left

Once the startup timeout expires, neither an overflow nor a cyclic restart of the timer is performed. The timer status is kept for further processing by the startup state machine.

Startup Noise Timeout

At the same time the startup timer is started for the first time (transition from STARTUP_PREPARE state to COLDSTART_LISTEN state), the startup noise timer is started. This additional timeout is used to improve reliability of the startup procedure in the presence of noise. The startup noise timeout is configured by programming FRSUCC2.LTN[3:0] (see section 19.8.2, FlexRay SUC Configuration Register 2 (FRSUCC2)).

The startup noise timeout is: pdListenTimeout × gListenNoise = FRSUCC2.LT[20:0] × (FRSUCC2.LTN[3:0] + 1)

The startup noise timer is restarted upon:

- Entering the COLDSTART_LISTEN state
- Reception of correctly decoded headers or CAS symbols while the node is in COLDSTART_LISTEN state

The startup noise timer is stopped when the COLDSTART_LISTEN state is left.

Once the startup noise timeout expires, neither an overflow nor a cyclic restart of the timer is performed. The status is kept for further processing by the startup state machine. Since the startup noise timer won't be restarted when random channel activity is sensed, this timeout defines the fall-back solution that guarantees that a node will try to start up the communication cluster even in the presence of noise.

(3) Path of Leading Coldstart Node (Initiating Coldstart)

When a coldstart node enters COLDSTART_LISTEN, it listens to its attached channels.

If no communication is detected, the node enters the COLDSTART_COLLISION_RESOLUTION state and commences a coldstart attempt. The initial transmission of a CAS symbol is succeeded by the first regular cycle. This cycle has the number zero.

From cycle zero on, the node transmits its startup frame. Since each coldstart node may perform a coldstart attempt, it may occur that several nodes simultaneously transmit the CAS symbol and enter the coldstart path. This situation is resolved during the first four cycles after CAS transmission.

As soon as a node that initiates a coldstart attempt receives a CAS symbol or a frame header during these four cycles, it re-enters the COLDSTART_LISTEN state. Thereby, only one node remains in this path. In cycle four, other coldstart nodes begin to transmit their startup frames.

After four cycles in COLDSTART_COLLISION_RESOLUTION state, the node that initiated the coldstart enters the COLDSTART_CONSISTENCY_CHECK state. It collects all startup frames from cycle four and five and performs the clock correction. If the clock correction does not deliver any errors and it has received at least one valid startup frame pair, the node leaves COLDSTART_CONSISTENCY_CHECK and enters NORMAL_ACTIVE state.

The number of coldstart attempts that a node is allowed to perform is configured by FRSUCC1.CSA[4:0]. The number of remaining coldstart attempts can be read from FRCCSV.RCA[4:0]. The number of remaining coldstart attempts is reduced by one for each attempted coldstart. A node may enter the COLDSTART_LISTEN state only if this value is larger than one and it may enter the COLDSTART_COLLISION_RESOLUTION state only if this value is larger than zero. If the number of coldstart attempts is one, coldstart is inhibited but integration is still possible.

(4) Path of Following Coldstart Node (Responding to Leading Coldstart Node)

When a coldstart node enters the COLDSTART_LISTEN state, it tries to receive a valid pair of startup frames to derive its schedule and clock correction from the leading coldstart node.

As soon as a valid startup frame has been received the INITIALIZE_SCHEDULE state is entered. If the clock synchronization can successfully receive a matching second valid startup frame and derive a schedule from this, the INTEGRATION_COLDSTART_CHECK state is entered.

In INTEGRATION_COLDSTART_CHECK state it is assured that the clock correction can be performed correctly and that the coldstart node from which this node has initialized its schedule is still available. The node collects all sync frames and performs clock correction in the following double-cycle. If clock correction does not signal any errors and if the node continues to receive sufficient frames from the same node it has integrated on, the COLDSTART_JOIN state is entered.

In COLDSTART_JOIN state following coldstart nodes begin to transmit their own startup frames and continue to do so in subsequent cycles. Thereby, the leading coldstart node and the nodes joining it can check if their schedules agree with each other. If the clock correction signals any error, the node aborts the integration attempt. If a node in this state sees at least one valid startup frame during all even cycles in this state and at least one valid startup frame pair during all double cycles in this state, the node leaves COLDSTART_JOIN state and enters NORMAL_ACTIVE state. Thereby it leaves STARTUP at least one cycle after the node that initiated the coldstart.

(5) Path of Non-Coldstart Node

When a non-coldstart node enters the INTEGRATION_LISTEN state, it listens to its attached channels.

As soon as a valid startup frame has been received, the INITIALIZE_SCHEDULE state is entered. If the clock synchronization can successfully receive a matching second valid startup frame and derive a schedule from this, the INTEGRATION_CONSISTENCY_CHECK state is entered.

In INTEGRATION_CONSISTENCY_CHECK state the node verifies that the clock correction can be performed correctly and that enough coldstart nodes (at least 2) are sending startup frames that agree with the node's own schedule. Clock correction is activated, and if any errors are signalled, the integration attempt is aborted.

During the first even cycle in this state, either two valid startup frames or the startup frame of the node that this node has integrated on must be received; otherwise the node aborts the integration attempt.

During the first double-cycle in this state, either two valid startup frame pairs or the startup frame pair of the node that this node has integrated on must be received; otherwise the node aborts the integration attempt.

If after the first double-cycle less than two valid startup frames are received within an even cycle, or less than two valid startup frame pairs are received within a double-cycle, the startup attempt is aborted.

Nodes in this state need to see two valid startup frame pairs for two consecutive double-cycles each to be allowed to leave STARTUP and enter NORMAL_OPERATION. Consequently, they leave startup at least one double-cycle after the node that initiated the coldstart and only at the end of a cycle with an odd cycle number.

19.18.7 NORMAL_ACTIVE State

As soon as the node that transmitted the first CAS symbol (resolving the potential access conflict and entering STARTUP via coldstart path) and one additional node have entered the NORMAL_ACTIVE state, the startup phase for the cluster has finished. In the NORMAL_ACTIVE state, all configured messages are scheduled for transmission. This includes all data frames as well as the sync frames. Rate and offset measurement is started in all even cycles (even/odd cycle pairs required).

In NORMAL_ACTIVE state the CC supports regular communication functions:

- The CC performs transmissions and reception on the FlexRay bus as configured
- Clock synchronization is running
- The Host interface is operational

The CC exits from that state to

- HALT state by writing FRSUCC1.CMD[3:0] = "0110" (HALT command, at the end of the current cycle)
- HALT state by writing FRSUCC1.CMD[3:0] = "0111" (FREEZE command, immediately)
- HALT state due to change of the error state from ACTIVE to COMM_HALT
- NORMAL_PASSIVE state due to change of the error state from ACTIVE to PASSIVE
- READY state by writing FRSUCC1.CMD[3:0] = "0010" (READY command)

19.18.8 NORMAL_PASSIVE State

NORMAL_PASSIVE state is entered from NORMAL_ACTIVE state when the error state changes from ACTIVE to PASSIVE.

In NORMAL_PASSIVE state, the node is able to receive all frames (node is fully synchronized and performs clock synchronization). Contrary to the NORMAL_ACTIVE state, the node does not actively participate in communication, i.e. neither symbols nor frames are transmitted.

In NORMAL_PASSIVE state

- The CC performs reception on the FlexRay bus
- The CC does not transmit any frames or symbols on the FlexRay bus
- Clock synchronization is running
- The Host interface is operational

The CC exits from this state to

- HALT state by writing FRSUCC1.CMD[3:0] = "0110" (HALT command, at the end of the current cycle)
- HALT state by writing FRSUCC1.CMD[3:0] = "0111" (FREEZE command, immediately)
- HALT state due to change of the error state from PASSIVE to COMM_HALT
- NORMAL_ACTIVE state due to change of the error state from PASSIVE to ACTIVE. The transition takes place when FRCCEV.PTAC[4:0] equals FRSUCC1.PTA[4:0] - 1
- To READY state by writing FRSUCC1.CMD[3:0] = "0010" (READY command)

19.18.9 HALT State

In this state all communication (reception and transmission) is stopped.

The CC enters this state

- By writing FRSUCC1.CMD[3:0] = "0110" (HALT command) while the CC is in NORMAL_ACTIVE or NORMAL_PASSIVE state
- By writing FRSUCC1.CMD[3:0] = "0111" (FREEZE command) from all states
- When exiting from NORMAL_ACTIVE state because the clock correction failed counter reached the "maximum without clock correction fatal" limit and the HCSE bit in the FRSUCC1.register is set to 1.
- When exiting from NORMAL_PASSIVE state because the clock correction failed counter reached the "maximum without clock correction fatal" limit and the HCSE bit in the FRSUCC1.register is set to 1.

The CC exits from this state to DEFAULT_CONFIG state.

- By writing FRSUCC1.CMD[3:0] = "0001" (CONFIG command)

When the CC enters HALT state, all configuration and status data is maintained for analysing purposes.

When the Host writes FRSUCC1.CMD[3:0] = "0110" (HALT command), the CC sets bit FRCCSV.HRQ and enters HALT state at the next end of cycle.

When the Host writes FRSUCC1.CMD[3:0] = "0111" (FREEZE command), the CC enters HALT state immediately and sets bit FRCCSV.FSI.

The POC state from which the transition to HALT state took place can be read from FRCCSV.PSL[5:0].

19.19 Network Management

The accrued Network Management (NM) vector can be read from registers FRNMV1 to FRNMV3. The CC performs a bit-wise OR operation over all NM vectors out of all received valid NM frames with the Payload Preamble Indicator (PPI) bit set. Only static frames may be configured to hold NM information. The CC updates the NM vector at the end of each cycle.

The length of the NM vector can be configured from 0 to 12 bytes by FRNEMC.NML[3:0]. The NM vector length must be configured identically in all nodes of a cluster.

To configure a transmit buffer to send FlexRay frames with the PPI bit set, bit PPIT in the header section of the respective transmit buffer has to be set via FRWRHS1.PPIT. In addition the Host has to write the NM information to the data section of the respective transmit buffer.

The evaluation of the NM vector has to be done by the application running on the Host.

Note: In case a message buffer is configured for transmission/reception of network management frames, the payload length configured in header 2 of that message buffer should be equal or greater than the length of the NM vector configured by FRNEMC.NML[3:0].
When the CC transits to HALT state, the cycle count is not incremented and therefore the NM vector is not updated. In this case FRNMV1 to FRNMV3 holds the value from the cycle before.

19.20 Filtering and Masking

Filtering is done by comparison of the configuration of assigned message buffers against actual slot and cycle counter values and channel ID (channel A, B). A message buffer is only updated/transmitted if the required matches occur.

Filtering is done on:

- Slot Counter
- Cycle Counter
- Channel ID

The following filter combinations for acceptance/transmit filtering are allowed:

- Slot Counter + Channel ID
- Slot Counter + Cycle Counter + Channel ID

All configured filters must match in order to store a received message in a message buffer.

Note: For the FIFO the acceptance filter is configured by the FlexRay FIFO Rejection Filter and the FlexRay FIFO Rejection Filter Mask.

A message will be transmitted in the time slot corresponding to the configured frame ID on the configured channel(s). If cycle counter filtering is enabled the configured cycle filter value must also match.

19.20.1 Slot Counter Filtering

Every transmit and receive buffer contains a frame ID stored in the header section. This frame ID is compared against the actual slot counter value in order to assign receive and transmit buffers to the corresponding slot.

If two or more message buffers are configured with the same frame ID, and if they have a matching cycle counter filter value for the same slot, then the message buffer with the lowest message buffer number is used.

19.20.2 Cycle Counter Filtering

Cycle counter filtering is based on the notion of a cycle set. For filtering purposes, a match is detected if any one of the elements of the cycle set is matched. The cycle set is defined by the cycle code field in header section 1 of each message buffer.

If message buffer 0 resp. 1 is configured to hold the startup/sync frame or the single slot frame by bits FRSUCC1.TXST, FRSUCC1.TXSY, and FRSUCC1.TSM, cycle counter filtering for message buffer 0 resp. 1 must be disabled.

Note: Sharing of a static time slot via cycle counter filtering between different nodes of a FlexRay network is not allowed.

The set of cycle numbers belonging to a cycle set is determined as described in table 19.8.

Table 19.8 Definition of Cycle Set

Cycle Code	Matching Cycle Counter Values
0b000000x	All cycles
0b000001c	Every second cycle When the residue of dividing the cycle count value by 2 is c
0b00001cc	Every fourth cycle When the residue of dividing the cycle count value by 4 is cc
0b0001ccc	Every eighth cycle When the residue of dividing the cycle count value by 8 is ccc
0b001cccc	Every sixteenth cycle When the residue of dividing the cycle count value by 16 is cccc
0b01ccccc	Every thirty-second cycle When the residue of dividing the cycle count value by 32 is ccccc
0b1cccccc	Every sixty-fourth cycle When the residue of dividing the cycle count value by 64 is cccccc

Table 19.9 below gives some examples for valid cycle sets to be used for cycle counter filtering.

Table 19.9 Examples for Valid Cycle Sets

Cycle Code	Matching Cycle Counter Values
0b0000011	1-3-5-7-.... -63 ↵
0b0000100	0-4-8-12-.... -60 ↵
0b0001110	6-14-22-30-.... -62 ↵
0b0011000	8-24-40-56 ↵
0b0100011	3-35 ↵
0b1001001	9 ↵

The received message is stored only if the cycle counter value of the cycle during which the message is received matches an element of the receive buffer's cycle set. Other filter criteria must also be met.

The content of a transmit buffer is transmitted on the configured channel(s) when an element of the cycle set matches the current cycle counter value. Other filter criteria must also be met.

19.20.3 Channel ID Filtering

There is a 2-bit channel filtering field (CHA, CHB) located in the header section of each message buffer in the Message RAM. It serves as a filter for receive buffers, and as a control field for transmit buffers (see table 19.10).

Table 19.10 Channel Filtering Configuration

CHA	CHB	Transmitting Frame in Transmit Buffer	Storing Valid Receive Frame in Receive Buffer
1	1	Both channels (static segment only)	Received on channel A or B (store first semantically valid frame, static segment only)
1	0	Channel A	Received on channel A
0	1	Channel B	Received on channel B
0	0	No transmission	Frame ignored

The contents of a transmit buffer is transmitted on the channels specified in the channel filtering field when the slot counter filtering and cycle counter filtering criteria are also met. Only in static segment a transmit buffer may be set up for transmission on both channels (CHA and CHB set).

Valid received frames are stored if they are received on the channels specified in the channel filtering field when the slot counter filtering and cycle counter filtering criteria are also met. Only in static segment a receive buffer may be setup for reception on both channels (CHA and CHB set).

Note: If a message buffer is configured for the dynamic segment and both bits of the channel filtering field are set to '1', no frames are transmitted resp. received frames are ignored (same function as CHA = CHB = '0').

19.20.4 FIFO Filtering

For FIFO filtering there is one rejection filter and one rejection filter mask available. The FIFO filter consists of channel filter FRFRF.CH[1:0], frame ID filter FRFRF.FID[10:0], and cycle counter filter FRFRF.CYF[6:0]. Registers FRFRF and FRFRFM can be configured in DEFAULT_CONFIG or CONFIG state only. The filter configuration in the header section of message buffers belonging to the FIFO is ignored.

The 7-bit cycle counter filter determines the cycle set to which frame ID and channel rejection filter are applied. In cycles not belonging to the cycle set specified by FRFRF.CYF[6:0], all frames are rejected.

A valid received frame is stored in the FIFO if channel ID, frame ID, and cycle counter are not rejected by the configured rejection filter and rejection filter mask, and if there is no matching dedicated receive buffer.

19.21 Transmit Process

19.21.1 Static Segment

For the static segment, if there are several messages pending for transmission, the message with the frame ID corresponding to the next sending slot is selected for transmission.

The data section of transmit buffers assigned to the static segment can be updated until the end of the preceding time slot. This means that a transfer from the Input Buffer has to be started by writing to the FlexRay Input Buffer Command Request register latest at this time.

19.21.2 Dynamic Segment

In the dynamic segment, if several messages are pending, the message with the highest priority (lowest frame ID) is selected next. In the dynamic segment different slot counter sequences on channel A and channel B are possible (concurrent sending of different frame IDs on both channels).

The data section of transmit buffers assigned to the dynamic segment can be updated until the end of the preceding slot. This means that a transfer from the Input Buffer has to be started by writing to the FlexRay Input Buffer Command Request register latest at this time.

The start of latest transmit configured by FRMHDC.SLT[12:0] defines the maximum minislot value allowed before inhibiting new frame transmission in the dynamic segment of the current cycle.

19.21.3 Transmit Buffers

FlexRay message buffers can be configured as transmit buffers by programming bit CFG in the header section of the respective message buffer to '1' via FRWRHS1.

There exists the following possibilities to assign a transmit buffer to the CC channels:

• Static segment:	Channel A or channel B
	Channel A and channel B
• Dynamic segment:	Channel A or channel B

Message buffer 0 resp. 1 is dedicated to hold the startup frame, the sync frame, or the designated single slot frame as configured by FRSUCC1.TXST, FRSUCC1.TXSY, and FRSUCC1.TSM. In this case, it can be reconfigured in DEFAULT_CONFIG or CONFIG state only. This ensures that any node transmits at most one startup/sync frame per communication cycle. Transmission of startup/sync frames from other message buffers is not possible.

All other message buffers configured for transmission in static or dynamic segment are reconfigurable during runtime depending on the configuration of FRMRC.SEC[1:0] (see section 19.24.1, Reconfiguration of Message Buffers). Due to the organization of the data partition in the Message RAM (reference by data pointer), reconfiguration of the configured payload length and the data pointer in the header section of a message buffer may lead to erroneous configurations.

If a message buffer is reconfigured (header section updated) during runtime, it may happen that this message buffer is not send out in the respective communication cycle.

The CC does not have the capability to calculate the header CRC. The Host is supposed to provide the header CRCs for all transmit buffers. If network management is required, the Host has to set the PPIT bit in the header section of the respective message buffer to '1' and write the network management information to the data section of the message buffer (see section 19.19, Network Management).

The payload length field configures the payload length in 2-byte words. If the configured payload length of a static transmit buffer is shorter than the payload length configured for the static segment by FRMHDC.SFDL[6:0], the CC generates padding bytes to ensure that frames have proper physical length. The padding pattern is logical zero.

Note: In case of an odd payload length (PLC = 1,3,5,...), the application has to write zero to the last 16 bits of the message buffers data section to ensure that the padding pattern is all zero.

Each transmit buffer provides a transmission mode flag TXM that allows the Host to configure the transmission mode for the transmit buffer. If this bit is set, the transmitter operates in the single-shot mode. If this bit is cleared, the transmitter operates in the continuous mode.

In single-shot mode the CC resets the respective TXR flag after transmission has completed. Now the Host may update the transmit buffer.

In continuous mode, the CC does not reset the respective transmission request flag TXR after successful transmission. In this case a frame is sent out each time the filter criteria match. The TXR flag can be reset by the Host by writing the respective message buffer number to the FRIBCR register while bit FRIBCM.STXRH is set to '0'.

If two or more transmit buffers meet the filter criteria simultaneously, the transmit buffer with the lowest message buffer number will be transmitted in the respective slot.

19.21.4 Frame Transmission

The following steps are required to prepare a message buffer for transmission:

- Configure the transmit buffer in the Message RAM via FRWRHS1, FRWRHS2, and FRWRHS3
- Write the data section of the transmit buffer via FRWRDSn
- Transfer the configuration and message data from Input Buffer to the Message RAM by writing the number of the target message buffer to register FRIBCR
- If configured in register FRIBCM, the transmission request flag TXR for the respective message buffer will be set as soon as the transfer has completed, and the message buffer is ready for transmission.
- Check whether the message buffer has been transmitted by checking the respective TXR bit (TXR = '0') in the TRXQ1/2/3/4 registers (single-shot mode only).

After transmission has completed, the respective TXR flag in the FRTXRQ1/2/3/4 register is reset (single-shot mode), and, if bit MBI in the header section of the message buffer is set, flag FRSIR.TXI is set to '1'. If enabled, an interrupt is generated.

19.21.5 Null Frame Transmission

If in static segment the Host does not set the transmission request flag before transmit time, and if there is no other transmit buffer with matching filter criteria, the CC transmits a null frame with the null frame indication bit set to '0' and the payload data set to zero.

In the following cases the CC transmits a null frame:

- If the message buffer with the lowest message buffer number matching the filter criteria does not have its transmission request flag set (TXR = '0').
- No transmit buffer configured for the slot has a cycle counter filter that matches the current cycle. In this case, no FlexRay message buffer status FRMBS is updated.

Null frames are not transmitted in the dynamic segment.

19.22 Receive Process

19.22.1 Dedicated Receive Buffers

A portion of the FlexRay message buffers can be configured as dedicated receive buffers by programming bit CFG in the header section of the respective message buffer to '0' via FRWRHS1.

The following possibilities exist to assign a receive buffer to the CC channels:

- Static segment: Channel A or channel B
  Channel A and channel B (the CC stores the first semantically valid frame)
- Dynamic segment: Channel A or channel B

The CC transfers the payload data of valid received messages from the shift register of the FlexRay channel protocol controller (channel A or B) to the receive buffer with the matching filter configuration. A receive buffer stores all frame elements except the frame CRC.

All message buffers configured for reception in static or dynamic segment are reconfigurable during runtime depending on the configuration of FRMRC.SEC[1:0] (see section 19.24.1, Reconfiguration of Message Buffers). If a message buffer is reconfigured (header section updated) during runtime it may happen that in the respective communication cycle a received message is lost.

If two or more receive buffers meet the filter criteria simultaneously, the receive buffer with the lowest message buffer number is updated with the received message.

19.22.2 Frame Reception

The following steps are required to prepare a dedicated message buffer for reception:

- Configure the receive buffer in the Message RAM via FRWRHS1, FRWRHS2, and FRWRHS3
- Transfer the configuration from Input Buffer to the Message RAM by writing the number of the target message buffer to register FRIBCR

Once these steps are performed, the message buffer functions as an active receive buffer and participates in the internal acceptance filtering process which takes place every time the CC receives a message. The first matching receive buffer is updated from the received message.

If a valid payload segment was stored in the data section of a message buffer, the respective ND flag in the FRNDAT1/2/3/4 registers is set, and, if bit MBI in the header section of that message buffer is set, flag FRSIR.RXI is set to '1'. If enabled, an interrupt is generated.

In case that bit ND was already set when the Message Handler updates the message buffer, bit FRMBS.MLST of the respective message buffer is set and the unprocessed message data is lost.

If no frame, a null frame, or a corrupted frame was received in a slot, the data section of the message buffer configured for this slot is not updated. In this case only the respective FlexRay message buffer status FRMBS is updated.

When the Message Handler changed the FlexRay message buffer status FRMBS in the header section of a message buffer, the respective MBC flag in the FRMBSC1/2/3/4 registers is set, and if bit MBI in the header section of that message buffer is set, flag FRSIR.MBSI is set to '1'. If enabled an interrupt is generated.

If the payload length of a received frame PLR[6:0] is longer than the value programmed by PLC[6:0] in the header section of the respective message buffer, the data field stored in the message buffer is truncated to that length.

To read a receive buffer from the Message RAM via the Output Buffer, proceed as described in section 19.24.2 (2), Data Transfer from Message RAM to Output Buffer.

Note: The ND and MBC flags are automatically cleared by the Message Handler when the payload data and the header of a received message have been transferred to the Output Buffer, respectively.

19.22.3 Null Frame Reception

The payload segment of a received null frame is not copied into the matching dedicated receive buffer. If a null frame has been received, only the FlexRay message buffer status FRMBS of the matching message buffer is updated from the received null frame. All bits in header 2 and 3 of the matching message buffer remain unchanged. They are updated from received data frames only.

When the Message Handler changed the FlexRay message buffer status FRMBS in the header section of a message buffer, the respective MBC flag in the FRMBSC1/2/3/4 register is set, and if bit MBI in the header section of that message buffer is set, flag FRSIR.MBSI is set to '1'. If enabled, an interrupt is generated.

19.22.4 Received Data Transfer to On-Chip RAM by A-DMAC

After the initial settings for starting the A-DMAC have been made and receive data is stored in the receive message buffer or receive FIFO among the FlexRay message buffers, the A-DMAC is activated and it transfers data in the receive message buffer to on-chip RAM. For the A-DMAC settings, see section 12, Automotive Direct Memory Access Controller (A-DMAC).

19.23 FIFO Function

19.23.1 Description

A group of the message buffers can be configured as a cyclic First-In-First-Out (FIFO) buffer. The group of message buffers belonging to the FIFO is contiguous in the register map starting with the message buffer referenced by FRMRC.FFB[7:0] and ending with the message buffer referenced by FRMRC.LCB[7:0]. Up to 127 message buffers can be assigned to the FIFO.

Every valid incoming message not matching with any dedicated receive buffer but passing the programmable FIFO filter is stored into the FIFO. In this case frame ID, payload length, receive cycle count, and the FlexRay message buffer status FRMBS of the addressed FIFO message buffer are overwritten with frame ID, payload length, receive cycle count, and the status from the received frame. Bit FRSIR.RFNE shows that the FIFO is not empty, bit FRSIR.RFCL is set when the receive FIFO fill level FRFSR.RFFL[7:0] is equal or greater than the critical level as configured by FRFCL.CL[7:0], bit FREIR.RFO shows that a FIFO overrun has been detected. If enabled, interrupts are generated.

If null frames are not rejected by the FlexRay FIFO rejection filter, the null frames will be treated like data frames when they are stored into the FIFO.

There are two index registers associated with the FIFO. The PUT Index Register (PIDX) is an index to the next available location in the FIFO. When a new message has been received it is written into the message buffer addressed by the PIDX register. The PIDX register is then incremented and addresses the next available message buffer. If the PIDX register is incremented past the highest numbered message buffer of the FIFO, the PIDX register is loaded with the number of the first (lowest numbered) message buffer in the FIFO chain. The GET Index Register (GIDX) is used to address the next message buffer of the FIFO to be read. The GIDX register is incremented after transfer of the contents of a message buffer belonging to the FIFO to the Output Buffer. The PUT Index Register and the GET Index Register are not accessible by the Host.

The FIFO is completely filled when the PUT index (PIDX) reaches the value of the GET index (GIDX). When the next message is written to the FIFO before the oldest message has been read, both PUT index and GET index are incremented and the new message overwrites the oldest message in the FIFO. This will set FIFO overrun flag FREIR.RFO.

A FIFO non empty status is detected when the PUT index (PIDX) differs from the GET index (GIDX). In this case flag FRSIR.RFNE is set. This indicates that there is at least one received message in the FIFO. The FIFO empty, FIFO not empty, and the FIFO overrun states are explained in figure 19.8 for a three message buffer FIFO.

The programmable FlexRay FIFO Rejection Filter (FRFRF) defines a filter pattern for messages to be rejected. The FIFO filter consists of channel filter, frame ID filter, and cycle counter filter. If bit FRFRF.RSS is set to '1' (default), all messages received in the static segment are rejected by the FIFO. If bit FRFRF.RNF is set to '1' (default), received null frames are not stored in the FIFO.

The FlexRay FIFO Rejection Filter Mask (FRFRFM) specifies which bits of the frame ID filter in the FlexRay FIFO Rejection Filter register are marked 'don't care' for rejection filtering.

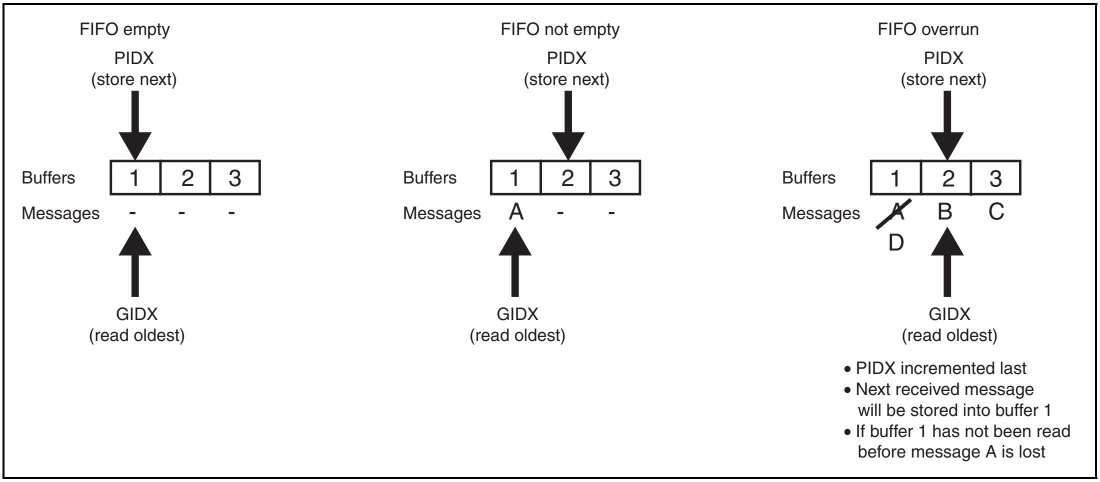


Figure 19.8 FIFO Status: Empty, Not Empty, Overrun

19.23.2 Configuration of the FIFO

(Re)configuration of message buffers belonging to the FIFO is only possible when the CC is in DEFAULT_CONFIG or CONFIG state. While the CC is in DEFAULT_CONFIG or CONFIG state, the FIFO function is not available.

For all message buffers belonging to the FIFO the payload length configured should be programmed to the same value via FRWRHS2.PLC[6:0]. The data pointer to the first 32-bit word of the data section of the respective message buffer in the Message RAM has to be configured via FRWRHS3.DP[10:0].

All information required for acceptance filtering is taken from the FlexRay FIFO rejection filter and the FlexRay FIFO rejection filter mask. The values configured in the header sections of the message buffers belonging to the FIFO are, with exception of DP and PLC, irrelevant.

Note: It is recommended to program the MBI bits of the message buffers belonging to the FIFO to '0' via FRWRHS1.MBI to avoid generation of RX interrupts.
If the payload length of a received frame is longer than the value programmed by FRWRHS2.PLC[6:0] in the header section of the respective message buffer, the data field stored in a message buffer of the FIFO is truncated to that length.

19.23.3 Access to the FIFO

For FIFO access outside DEFAULT_CONFIG and CONFIG state, the Host has to trigger a transfer from the Message RAM to the Output Buffer by writing the number of the first message buffer of the FIFO (referenced by FRMRC.FFB[7:0]) to the register FROBCR. The Message Handler then transfers the message buffer addressed by the GET Index Register (GIDX) to the Output Buffer. After this transfer the GET Index Register (GIDX) is incremented.

19.24 Message Handling

The Message Handler controls data transfers between the Input/Output Buffer and the Message RAM and between the Message RAM and the two Transient Buffer RAMs. All accesses to the internal RAMs are (32 + 1)-bit accesses. The additional bit is used for parity checking.

Access to the message buffers stored in the Message RAM is done under control of the Message Handler state machine. This avoids conflicts between accesses of the two FlexRay channel protocol controllers and the Host to the Message RAM.

Frame IDs of message buffers assigned to the static segment have to be in the range from 1 to FRGTUC7.NSS[9:0]. Frame IDs of message buffers assigned to the dynamic segment have to be in the range from FRGTUC7.NSS[9:0] + 1 to 2047.

Received messages with no matching dedicated receive buffer (static or dynamic segment) are stored in the receive FIFO (if configured) if they pass the FlexRay FIFO rejection filter.

19.24.1 Reconfiguration of Message Buffers

In case that an application needs to operate with more than 128 different messages, static and dynamic message buffers may be reconfigured during FlexRay operation. This is done by updating the header section of the respective message buffer via Input Buffer registers FRWRHS1 to FRWRHS3.

Reconfiguration has to be enabled via control bits FRMRC.SEC[1:0] in the FlexRay Message RAM Configuration register.

If a message buffer has not been transmitted/updated from a received frame before reconfiguration starts, the respective message is lost.

The point in time when a reconfigured message buffer is ready for transmission/reception according to the reconfigured frame ID depends on the actual state of the slot counter when the update of the header section has completed. Therefore it may happen that a reconfigured message buffer is not transmitted/updated from a received frame in the cycle where it was reconfigured.

The Message RAM is scanned according to table 19.11 below.

Table 19.11 Scan of Message RAM

Start of Scan	Scan for Slots
1	2...15, 1 (next cycle)
8	16...23, 1 (next cycle)
16	24...31, 1 (next cycle)
24	32...39, 1 (next cycle)
...	...

A Message RAM scan is terminated with the start of NIT regardless whether it has completed or not. The scan of the Message RAM for slots 2 to 15 starts at the beginning of slot 1 of the actual cycle. The scan of the Message RAM for slot 1 is done in the cycle before by checking in parallel to each scan of the Message RAM whether there is a message buffer configured for slot 1 of the next cycle.

The number of the first dynamic message buffer is configured by FRMRC.FDB[7:0]. In case a Message RAM scan starts while the CC is in dynamic segment, the scan starts with the message buffer number configured by FRMRC.FDB[7:0].

In case a message buffer should be reconfigured to be used in slot 1 of the next cycle, the following has to be considered:

- If the message buffer to be reconfigured for slot 1 is part of the "Static Buffers", it will only be found if it is reconfigured before the last Message RAM scan in the static segment of the actual cycle evaluates this message buffer.
- If the message buffer to be reconfigured for slot 1 is part of the "Static + Dynamic Buffers", it will be found if it is reconfigured before the last Message RAM scan in the actual cycle evaluates this message buffer.
- The start of NIT terminates the Message RAM scan. In case the Message RAM scan has not evaluated the reconfigured message buffer until this point in time, the message buffer will not be considered for the next cycle.

Note: Reconfiguration of message buffers may lead to the loss of messages and therefore has to be used very carefully. In worst case (reconfiguration in consecutive cycles) it may happen that a message buffer is never transmitted/updated from a received frame.

19.24.2 Host Access to Message RAM

The message transfer between Input Buffer and Message RAM as well as between Message RAM and Output Buffer is triggered by the Host by writing the number of the target/source message buffer to be accessed to FRIBCR or FROBCR register.

The FRIBCM and FROBCM registers can be used to write/read header and data section of the selected message buffer separately.

If bit FRIBCM.STXR is set to = '1', the transmission request flag TXR of the selected message buffer is automatically set after the message buffer has been updated. If bit FRIBCM.STXR is reset to '0', the transmission request flag TXR of the selected message buffer is reset. This can be used to stop transmission from message buffers operated in continuous mode.

Input Buffer (IBF) and Output Buffer (OBF) are build up as a double buffer structure. One half of this double buffer structure is accessible by the Host (IBF Host/OBF Host), while the other half (IBF Shadow/OBF Shadow) is accessed by the Message Handler for data transfers between IBF/OBF and Message RAM.

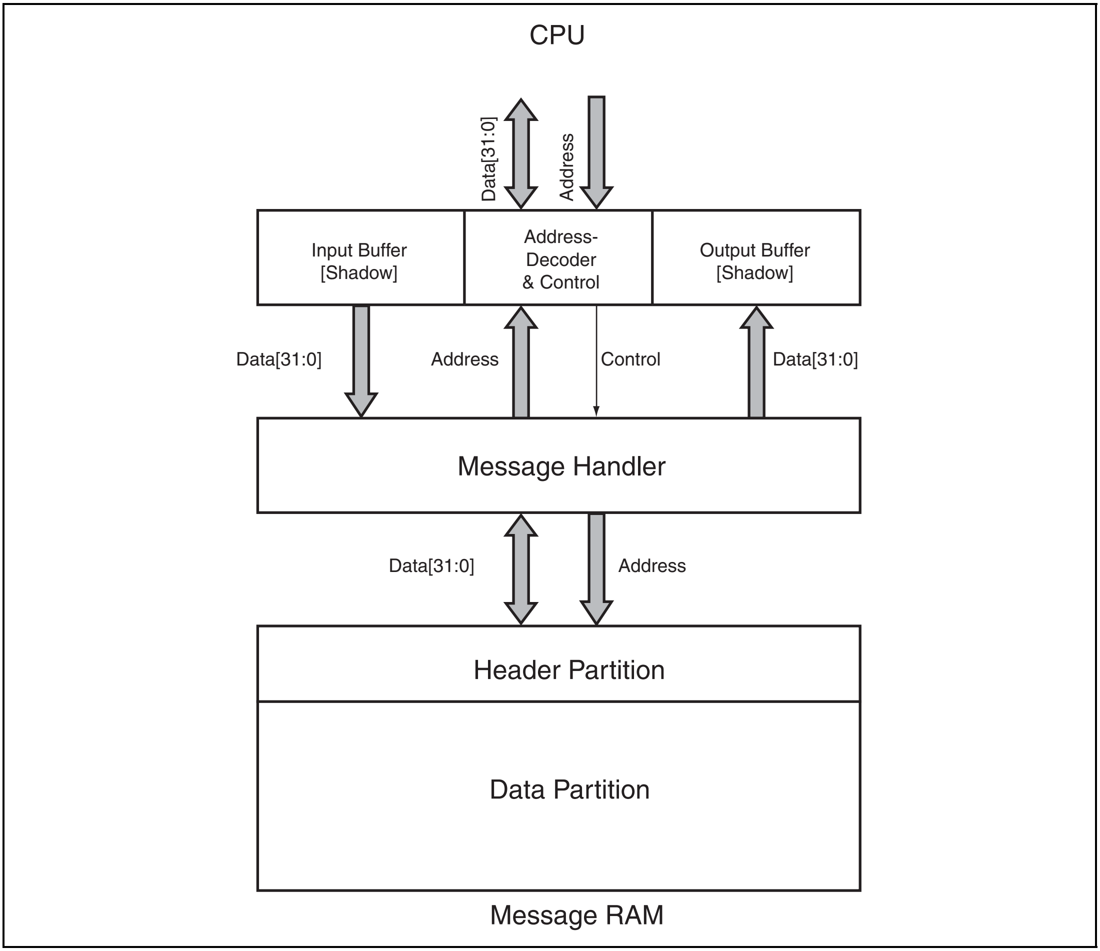


Figure 19.9 Host Access to Message RAM

(1) Data Transfer from Input Buffer to Message RAM

To configure/update a message buffer in the Message RAM, the Host has to write the data to FRWRDSn and the header to FRWRHS1 to FRWRHS3. The specific action is selected by configuring the FlexRay Input Buffer Command Mask FRIBCM.

When the Host writes the number of the target message buffer in the Message RAM to FRIBCR.IBRH[6:0], IBF Host and IBF Shadow are swapped (see figure 19.10).

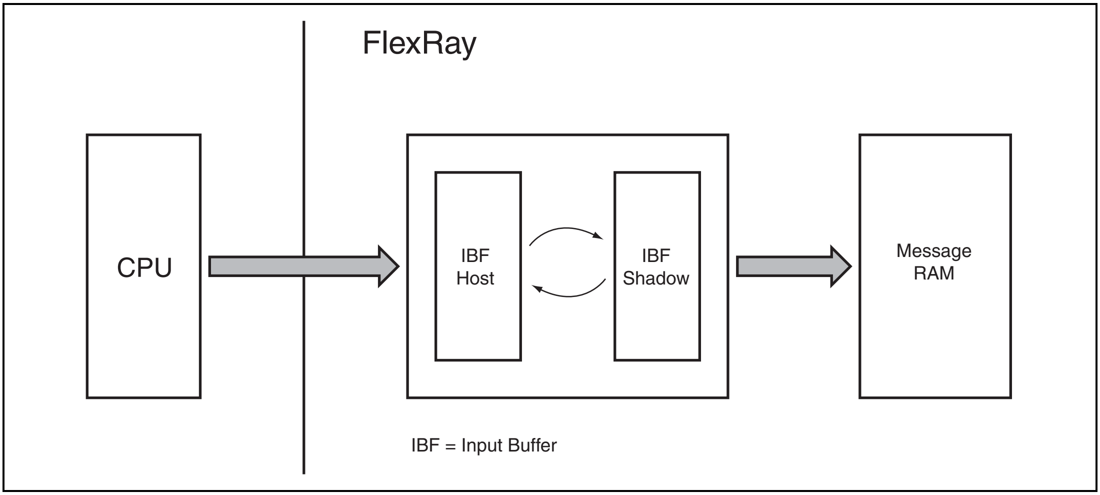


Figure 19.10 Double Buffer Structure of Input Buffer

In addition the bits in the FRIBCM and FRIBCR registers are also swapped to keep them attached to the respective IBF section (see figure 19.11).

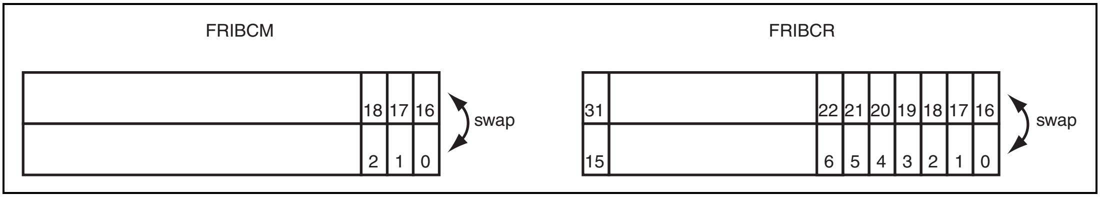


Figure 19.11 Swapping of FRIBCM and FRIBCR Bits

With this write operation bit FRIBCR.IBSYS is set to '1'. The Message Handler then starts to transfer the contents of IBF Shadow to the message buffer in the Message RAM selected by FRIBCR.IBRS[6:0].

While the Message Handler transfers the data from IBF Shadow to the target message buffer in the Message RAM, the Host may write the next message to IBF Host. After the transfer between IBF Shadow and the Message RAM has completed, bit FRIBCR.IBSYS is set back to '0' and the next transfer to the Message RAM may be started by the Host by writing the respective target message buffer number to FRIBCR.IBRH[6:0].

If a write access to FRIBCR.IBRH[6:0] occurs while FRIBCR.IBSYS is '1', FRIBCR.IBSYH is set to '1'. After completion of the ongoing data transfer from IBF Shadow to the Message RAM, IBF Host and IBF Shadow are swapped, FRIBCR.IBSYH is reset to '0', FRIBCR.IBSYS remains set to '1', and the next transfer to the Message RAM is started. In addition the message buffer numbers stored under FRIBCR.IBRH[6:0] and FRIBCR.IBRS[6:0] and the command mask flags are also swapped.

Example of Host Access to Message Buffers

Configure/update n-th message buffer via IBF:

- Wait until FRIBCR.IBSYH is reset
- Write data section to FRWRDSn
- Write header section to FRWRHS1 to FRWRHS3
- Write Command Mask: write FRIBCM.STXRH, FRIBCM.LDSH, FRIBCM.LHSH
- Demand data transfer to target message buffer: write FRIBCR.IBRH[6:0]

Configure/update (n+1)th message buffer via IBF:

- Wait until FRIBCR.IBSYH is reset
- Write data section to FRWRDSn
- Write header section to FRWRHS1 to FRWRHS3
- Write Command Mask: write FRIBCM.STXRH, FRIBCM.LDSH, FRIBCM.LHSH
- Demand data transfer to target message buffer: write FRIBCR.IBRH[6:0]

Note: Any write access to IBF while FRIBCR.IBSYH is '1' will set error flag FREIR.IIBA to '1'. In this case the write access has no effect.

Table 19.12 Assignment of FRIBCM Bits

Bit No.	Access	Bit	Function
18	r	STXRS	Set Transmission Request Shadow ongoing or finished
17	r	LDSS	Load Data Section Shadow ongoing or finished
16	r	LHSS	Load Header Section Shadow ongoing or finished
2	r/w	STXRH	Set Transmission Request Host
1	r/w	LDSH	Load Data Section Host
0	r/w	LHSH	Load Header Section Host

Table 19.13 Assignment of FRIBCR Bits

Bit No.	Access	Bit	Function
31	r	IBSYS	IBF Busy Shadow, signals ongoing transfer from IBF Shadow to Message RAM
22 to 16	r	IBRS[6:0]	IBF Request Shadow, number of message buffer currently/lately updated
15	r	IBSYH	IBF Busy Host, transfer request pending for message buffer referenced by IBRH[6:0]
6 to 0	r/w	IBRH[6:0]	IBF Request Host, number of message buffer to be updated next

(2) Data Transfer from Message RAM to Output Buffer

To read a message buffer from the Message RAM, the Host has to write to register FROBCR to trigger the data transfer as configured in FROBCM. After the transfer has completed, the Host can read the transferred data from FRRDDSn, FRRDHS1 to FRRDHS3, and FRMBS.

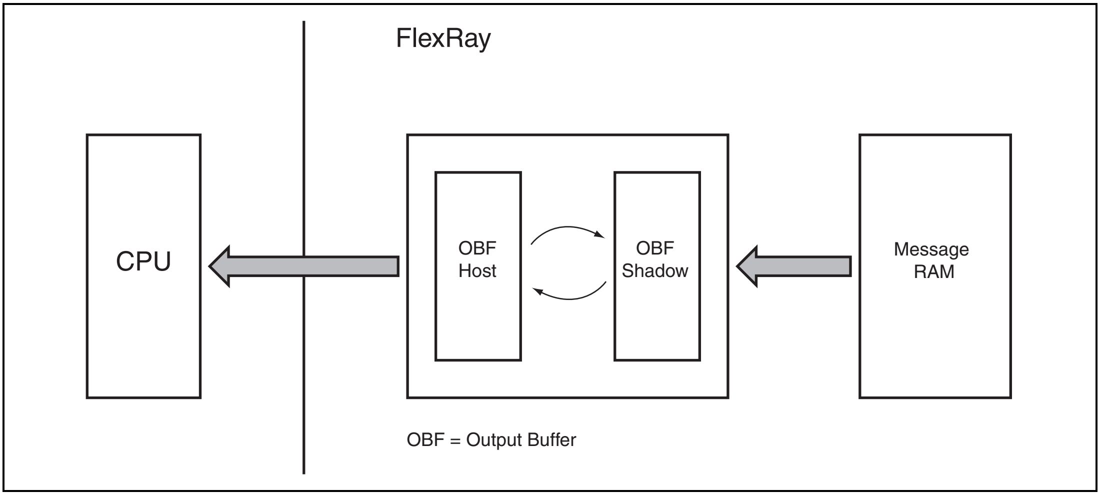


Figure 19.12 Double Buffer Structure of Output Buffer

OBF Host and OBF Shadow as well as bits FROBCM.RHSS, FROBCM.RDSS, FROBCM.RHSH, FROBCM.RDSH and bits FROBCR.OBRS[6:0], FROBCR.OBRH[6:0] are swapped under control of bits FROBCR.VIEW and FROBCR.REQ.

Writing bit FROBCR.REQ to '1' copies bits FROBCM.RHSS, FROBCM.RDSS and bits FROBCR.OBRS[6:0] to an internal storage (see figure 19.13).

After setting FROBCR.REQ to '1', FROBCR.OBSYS is set to '1', and the transfer of the message buffer selected by FROBCR.OBRS[6:0] from the Message RAM to OBF Shadow is started. After the transfer between the Message RAM and OBF Shadow has completed, the FROBCR.OBSYS bit is set back to '0'. Bits FROBCR.REQ and FROBCR.VIEW can only be set to '1' while FROBCR.OBSYS is '0'.

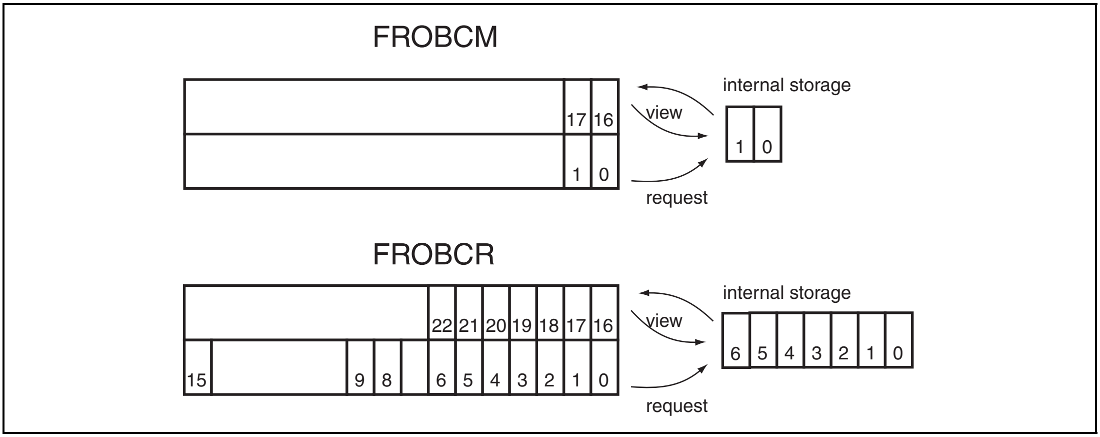


Figure 19.13 Swapping of FROBCM and FROBCR Bits

OBF Host and OBF Shadow are swapped by setting bit FROBCR.VIEW to '1' while bit FROBCR.OBSYS is '0' (see figure 19.12).

In addition bits FROBCR.OBRH[6:0] and bits FROBCM.RHSH, FROBCM.RDSH are swapped with the registers internal storage thus assuring that the message buffer number stored in FROBCR.OBRH[6:0] and the mask configuration stored in FROBCM.RHSH, FROBCM.RDSH matches the transferred data stored in OBF Host (see figure 19.13).

Now the Host can read the transferred message buffer from OBF Host while the Message Handler may transfer the next message from the Message RAM to OBF Shadow.

If bits REQ and VIEW are set to '1' with the same write access while OBSYS is '0', OBSYS is automatically set to '1' and OBF Shadow and OBF Host are swapped. Additionally mask bits FROBCM.RDSH and FROBCM.RHSH are swapped with the registers internal storage to keep them attached to the respective Output Buffer transfer. Afterwards OBRS[6:0] is copied to the register internal storage, mask bits FROBCM.RDSS and FROBCM.RHSS are copied to register OBCM internal storage, and the transfer of the selected message buffer from the Message RAM to OBF Shadow is started. While the transfer is ongoing the Host can read the message buffer transferred by the previous transfer from OBF Host. When the current transfer between Message RAM and OBF Shadow has completed, this is signaled by setting OBSYS back to '0'.

Example of Single Host Access to a Message Buffer

If a single message buffer has to be read out, two separate write accesses to FROBCR.REQ and FROBCR.VIEW are necessary.

- Wait until FROBCR.OBSYS is reset
- Write FlexRay Output Buffer Command Mask FROBCM.RHSS, FROBCM.RDSS
- Request transfer of message buffer to OBF Shadow by writing FROBCR.OBRS[6:0] and FROBCR.REQ (in case of and 8-bit Host interface, FROBCR.OBRS[6:0] has to be written before FROBCR.REQ).
- Wait until FROBCR.OBSYS is reset
- Toggle OBF Shadow and OBF Host by writing FROBCR.VIEW = '1'
- Read out transferred message buffer by reading FRRDDSn, FRRDHS1 to FRRDHS3, and FRMBS

Example of Continuous Host Access to Message Buffers

Request transfer of 1st message buffer to OBF Shadow:

- Wait until FROBCR.OBSYS is reset
- Write FlexRay Output Buffer Command Mask FROBCM.RHSS, FROBCM.RDSS for 1st message buffer
- Request transfer of 1st message buffer to OBF Shadow by writing FROBCR.OBRS[6:0] and FROBCR.REQ (in case of an 8-bit Host interface, FROBCR.OBRS[6:0] has to be written before FROBCR.REQ).

Toggle OBF Shadow and OBF Host to read out 1st transferred message buffer and request transfer of 2nd message buffer:

- Wait until FROBCR.OBSYS is reset
- Write FlexRay Output Buffer Command Mask FROBCM.RHSS, FROBCM.RDSS for 2nd message buffer
- Toggle OBF Shadow and OBF Host and start transfer of 2nd message buffer to OBF Shadow simultaneously by writing FROBCR.OBRS[6:0] of 2nd message buffer, FROBCR.REQ, and FROBCR.VIEW (in case of and 8-bit Host interface, FROBCR.OBRS[6:0] has to be written before FROBCR.REQ and FROBCR.VIEW).
- Read out 1st transferred message buffer by reading FRRDDSn, FRRDHS1 to FRRDHS3, and FRMBS

Demand access to last requested message buffer without request of another message buffer:

- Wait until FROBCR.OBSYS is reset
- Demand access to last transferred message buffer by writing FROBCR.VIEW
- Read out last transferred message buffer by reading FRRDDSn, FRRDHS1 to FRRDHS3, and FRMBS

Table 19.14 Assignment of FROBCM Bits

Bit No.	Access	Bit	Function
17	r	RDSH	Data Section available for Host access
16	r	RHSH	Header Section available for Host access
1	r/w	RDSS	Read Data Section Shadow
0	r/w	RHSS	Read Header Section Shadow

Table 19.15 Assignment of FROBCR Bits

Bit No.	Access	Bit	Function
22 to 16	r	OBRH[6:0]	OBF Request Host, number of message buffer available for Host access
15	r	OBSYS	OBF Busy Shadow, signals ongoing transfer from Message RAM to OBF Shadow
9	r/w	REQ	Request Transfer from Message RAM to OBF Shadow
8	r/w	VIEW	View OBF Shadow, swap OBF Shadow and OBF Host
6 to 0	r/w	OBRS[6:0]	OBF Request Shadow, number of message buffer for next request

19.24.3 FlexRay Protocol Controller Access to Message RAM

The two Transient Buffer RAMs (TBFA and TBFB) are used to buffer the data for transfer between the two FlexRay Protocol Controllers and the Message RAM.

Each Transient Buffer RAM is build up as a double buffer, able to store two complete FlexRay messages. There is always one buffer assigned to the corresponding Protocol Controller while the other one is accessible by the Message Handler.

If e.g. the Message Handler writes the next message to be send to Transient Buffer Tx, the FlexRay Channel Protocol Controller can access Transient Buffer Rx to store the message it is actually receiving. During transmission of the message stored in Transient Buffer Tx, the Message Handler transfers the last received message stored in Transient Buffer Rx to the Message RAM (if it passes acceptance filtering) and updates the respective message buffer.

Data transfers between the Transient Buffer RAMs and the shift registers of the FlexRay Channel Protocol Controllers are done in 32-bit units. This enables the use of a 32-bit shift register independent of the length of the FlexRay messages.

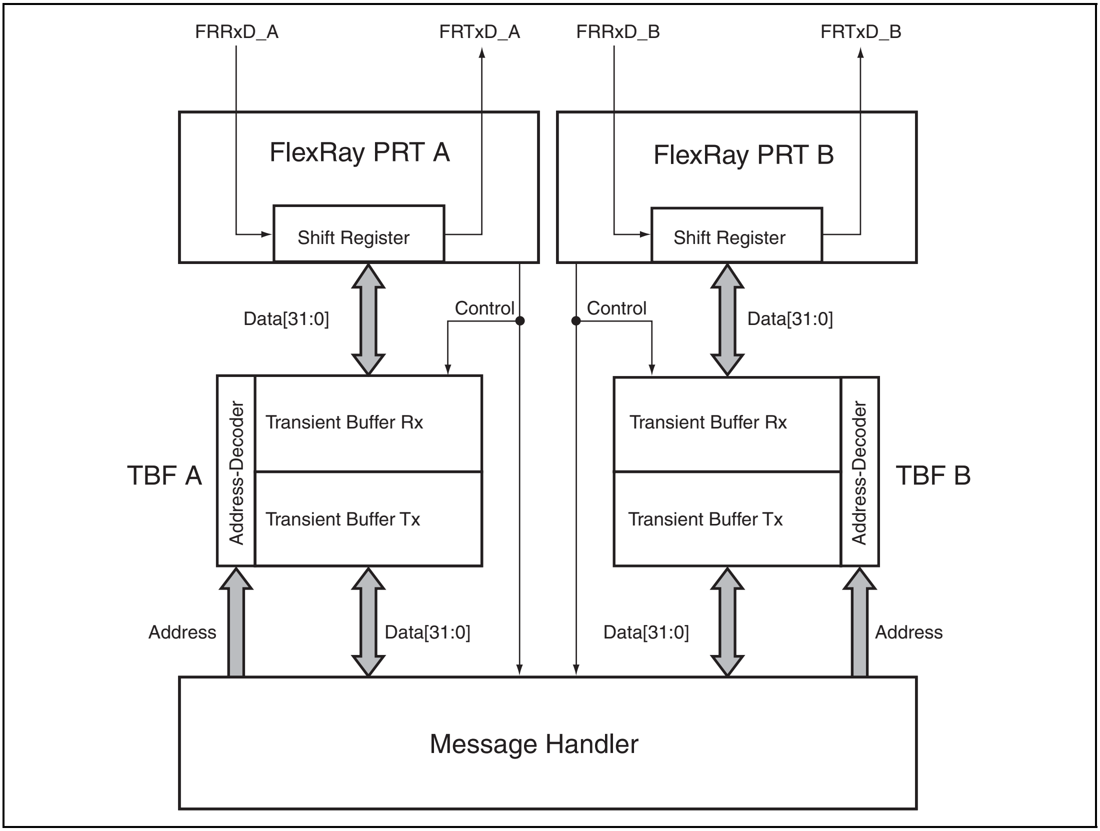


Figure 19.14 Access to Transient Buffer RAMs

19.25 Message RAM

To avoid conflicts between Host access to the Message RAM and FlexRay message reception/transmission, the Host cannot directly access the message buffers in the Message RAM. These accesses are handled via the Input and Output Buffers. The Message RAM is able to store up to 128 message buffers depending on the configured payload length.

The Message RAM is organized 2048 × 33 = 67,584 bits. Each 32-bit longword is protected by a parity bit. To achieve the required flexibility with respect to different numbers of data bytes per FlexRay frame (0 to 254), the Message RAM has a structure as shown in figure 19.15.

The data partition is allowed to start at Message RAM longword number: (FRMRC.LCB + 1) × 4.

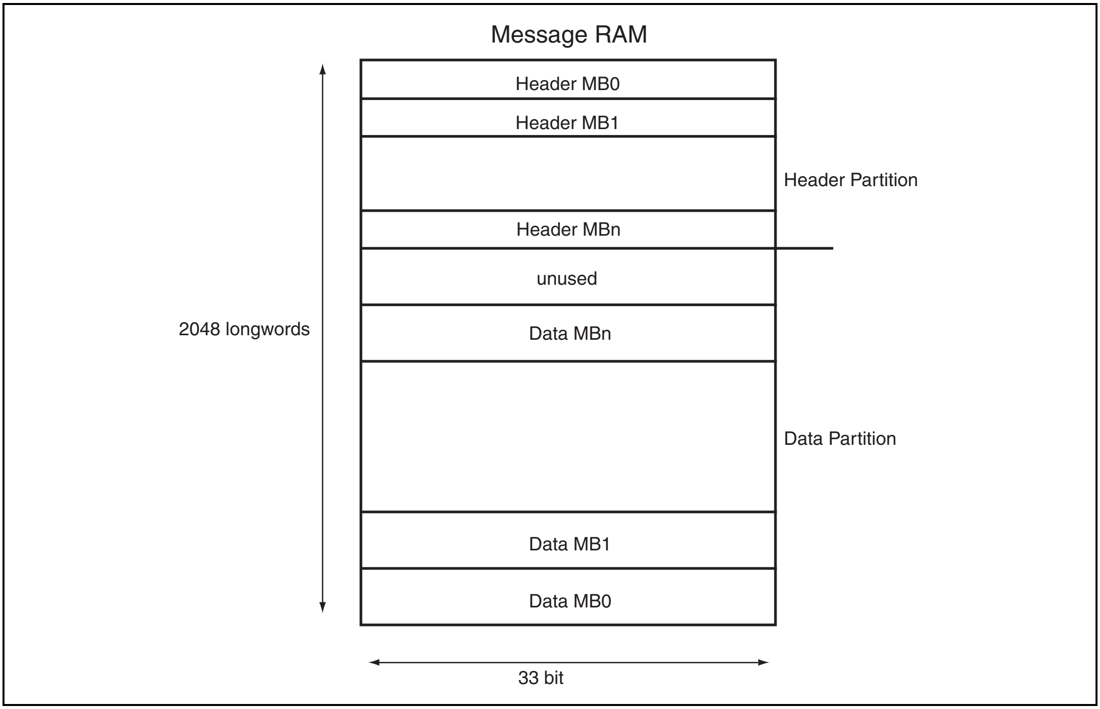


Figure 19.15 Configuration Example of Message Buffers in Message RAM

Header Partition

Stores header sections of the configured message buffers:

- Supports a maximum of 128 message buffers

- Each message buffer has a header section of four (32 + 1)-bit longwords
- Header 3 of each message buffer holds the 11-bit data pointer to the respective data section in the data partition

Data Partition

Flexible storage of data sections with different length. Some maximum values are:

- 30 message buffers with 254-byte data section each
- 56 message buffers with 128-byte data section each
- 128 message buffers with 48-byte data section each

Note: (header partition + data partition) should not occupy more than 2048 longwords.

19.25.1 Header Partition

The elements used for configuration of a message buffer as well as the actual message buffer status are stored in the header partition of the Message RAM as listed in table 19.16 below. Configuration of the header sections of the message buffers is done via IBF (FRWRHS1 to FRWRHS3). Read access to the header sections is done via OBF (FRRDHS1 to FRRDHS3 + FRMBS). The data pointer has to be calculated by the programmer to define the starting point of the data section for the respective message buffer in the data partition of the Message RAM. The data pointer should not be modified during runtime. For message buffers belonging to the receive FIFO (re)configuration is possible in DEFAULT_CONFIG or CONFIG state only.

The header section of each message buffer occupies four 33-bit words in the header partition of the Message RAM. The header of message buffer 0 starts with the first word in the Message RAM.

For transmit buffers the Header CRC has to be calculated by the Host.

Payload Length Received PLR[6:0], Receive Cycle Count RCC[5:0], Received on Channel Indicator RCI, Startup Frame Indicator SFI, Sync Frame Indicator SYN, Null Frame Indicator NFI, Payload Preamble Indicator PPI, and Reserved Bit RES are updated from received valid data frames only.

Header word 3 of each configured message buffer holds the respective FlexRay Message Buffer Status FRMBS.

Table 19.16 Header Section of Message Buffer in Message RAM

Bit Word	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	P			MBI	TXM	PPIT	CFG	CHB	CHA		Cycle Code												Frame ID										
1	P		Payload Length Received								Payload Length Configured												Tx Buffer: Header CRC Configured Rx Buffer: Header CRC Configured Received										
2	P			RES	PPI	NFI	SYN	SFI	RCI			Receive Cycle Count											Data Pointer										
3	P			RESS	PPIS	NFIS	SYNS	SFIS	RCIS			Cycle Count Status						FTB	FTA		MLST	ESB	ESA	TCIB	TCIA	SVOB	SVOA	CEOB	CEOA	SEOB	SEOA	VFRB	VFRA
•••	P	•••																															
•••	P	•••																															

- 🟥 Frame Configuration
- ⬜ Filter Configuration
- 🟦 Message Buffer Control
- 🟩 Message RAM Configuration
- 🟦 Updated from received Data Frame
- 🟪 Message Buffer Status FRMBS
- 🟨 Parity Bit
- ⬜ unused

Header 1 (Word 0)

Write access via FRWRHS1, read access via FRRDHS1:

- Frame ID - Slot counter filtering configuration
- Cycle Code - Cycle counter filtering configuration
- CHA, CHB - Channel filtering configuration
- CFG - Message buffer direction configuration: receive/transmit
- PPIT - Payload Preamble Indicator Transmit
- TXM - Transmit mode configuration: single-shot/continuous
- MBI - Message buffer receive/transmit interrupt enable

Header 2 (Word 1)

Write access via FRWRHS2, read access via FRRDHS2:

- Header CRC - Transmit Buffer: Configured by the Host (calculated from frame header)
  — Receive Buffer: Updated from received frame
- Payload Length Configured - Length of data section (2-byte words) as configured by the Host
- Payload Length Received - Length of payload segment (2-byte words) stored from received frame

Header 3 (Word 2)

Write access via FRWRHS3, read access via FRRDHS3:

- Data Pointer - Pointer to the beginning of the corresponding data section in the data partition

Read access via FRRDHS3, valid for receive buffers only, updated from received frames:

- Receive Cycle Count - Cycle count from received frame
- RCI - Received on Channel Indicator
- SFI - Startup Frame Indicator
- SYN - Sync Frame Indicator
- NFI - Null Frame Indicator
- PPI - Payload Preamble Indicator
- RES - Reserved bit

FlexRay Message Buffer Status FRMBS (Word 3)

Read access via FRMBS, updated by the CC at the end of the configured slot.

- VFRA - Valid Frame Received on channel A
- VFRB - Valid Frame Received on channel B
- SEOA - Syntax Error Observed on channel A
- SEOB - Syntax Error Observed on channel B
- CEOA - Content Error Observed on channel A
- CEOB - Content Error Observed on channel B
- SVOA - Slot boundary Violation Observed on channel A
- SVOB - Slot boundary Violation Observed on channel B
- TCIA - Transmission Conflict Indication channel A
- TCIB - Transmission Conflict Indication channel B

- ESA - Empty Slot Channel A
- ESB - Empty Slot Channel B
- MLST - Message LoST
- FTA - Frame Transmitted on Channel A
- FTB - Frame Transmitted on Channel B
- Cycle Count Status- Actual cycle count when status was updated
- RCIS - Received on Channel Indicator Status
- SFIS - Startup Frame Indicator Status
- SYNS - Sync Frame Indicator Status
- NFIS - Null Frame Indicator Status
- PPIS - Payload Preamble Indicator Status
- RESS - Reserved bit Status

19.25.2 Data Partition

The data partition of the Message RAM stores the data sections of the message buffers configured for reception/transmission as defined in the header partition. The number of data bytes for each message buffer can vary from 0 to 254. To optimize the data transfer between the shift registers of the two FlexRay Protocol Controllers and the Message RAM as well as between the Host interface and the Message RAM, the physical width of the Message RAM is set to 4 bytes plus one parity bit.

The data partition starts after the last word of the header partition. When configuring the message buffers in the Message RAM the programmer has to assure that the data pointers point to addresses within the data partition. Table 19.17 shows an example how the data sections of the configured message buffers can be stored in the data partition of the Message RAM.

The beginning and the end of a message buffer's data section is determined by the data pointer and the payload length configured in the message buffer's header section, respectively. This enables a flexible usage of the available RAM space for storage of message buffers with different data length.

If the size of the data section is an odd number of 2-byte words, the remaining 16 bits in the last 32-bit word are unused (see table 19.16).

Table 19.17 Example of Data Partition Structure in Message RAM

<table>
<tr><th>Bit
Word</th><th>32</th><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>...</td><td>P</td><td colspan="8">unused</td><td colspan="8">unused</td><td colspan="8">unused</td><td colspan="8">unused</td></tr>
<tr><td>...</td><td>P</td><td colspan="8">unused</td><td colspan="8">unused</td><td colspan="8">unused</td><td colspan="8">unused</td></tr>
<tr><td>...</td><td>P</td><td colspan="8">MBn Data3</td><td colspan="8">MBn Data2</td><td colspan="8">MBn Data1</td><td colspan="8">MBn Data0</td></tr>
<tr><td>...</td><td>P</td><td colspan="8">...</td><td colspan="8">...</td><td colspan="8">...</td><td colspan="8">...</td></tr>
<tr><td>...</td><td>P</td><td colspan="8">...</td><td colspan="8">...</td><td colspan="8">...</td><td colspan="8">...</td></tr>
<tr><td>...</td><td>P</td><td colspan="8">MBn Data(m)</td><td colspan="8">MBn Data(m-1)</td><td colspan="8">MBn Data(m-2)</td><td colspan="8">MBn Data(m-3)</td></tr>
<tr><td>...</td><td>P</td><td colspan="8">...</td><td colspan="8">...</td><td colspan="8">...</td><td colspan="8">...</td></tr>
<tr><td>...</td><td>P</td><td colspan="8">...</td><td colspan="8">...</td><td colspan="8">...</td><td colspan="8">...</td></tr>
<tr><td>...</td><td>P</td><td colspan="8">...</td><td colspan="8">...</td><td colspan="8">...</td><td colspan="8">...</td></tr>
<tr><td>...</td><td>P</td><td colspan="8">MB1 Data3</td><td colspan="8">MB1 Data2</td><td colspan="8">MB1 Data1</td><td colspan="8">MB1 Data0</td></tr>
<tr><td>...</td><td>P</td><td colspan="8">...</td><td colspan="8">...</td><td colspan="8">...</td><td colspan="8">...</td></tr>
<tr><td>...</td><td>P</td><td colspan="8">MB1 Data(k)</td><td colspan="8">MB1 Data(k-1)</td><td colspan="8">MB1 Data(k-2)</td><td colspan="8">MB1 Data(k-3)</td></tr>
<tr><td>2046</td><td>P</td><td colspan="8">MB0 Data3</td><td colspan="8">MB0 Data2</td><td colspan="8">MB0 Data1</td><td colspan="8">MB0 Data0</td></tr>
<tr><td>2047</td><td>P</td><td colspan="8">unused</td><td colspan="8">unused</td><td colspan="8">MB0 Data5</td><td colspan="8">MB0 Data4</td></tr>
</table>

19.25.3 Parity Check

There is a parity checking mechanism implemented in the FlexRay core to assure the integrity of the data stored in the seven RAM blocks. The RAM blocks have a parity generator/checker attached as shown in figure 19.16. When data is written to a RAM block, the local parity generator generates the parity bit. The FlexRay core uses an even parity (with an even number of ones in the 32-bit data word a zero parity bit is generated). The parity bit is stored together with the respective data word. The parity is checked each time a data word is read from any of the RAM blocks. The FlexRay core's internal data buses have a width of 32 bits.

If a parity error is detected, the respective error flag is set. The parity error flags FRMHDS.PIBF, FRMHDS.POBF, FRMHDS.PMR, FRMHDS.PTBF1, FRMHDS.PTBF2, and the faulty message buffer indicators FRMHDS.FMBD, FRMHDS.MFMB, FRMHDS.FMB[6:0] are located in the FlexRay Message Handler Status register. These single error flags control the FlexRay error interrupt flag FREIR.PERR.

Figure 19.16 shows the data paths between the RAM blocks and the parity generators/checkers.

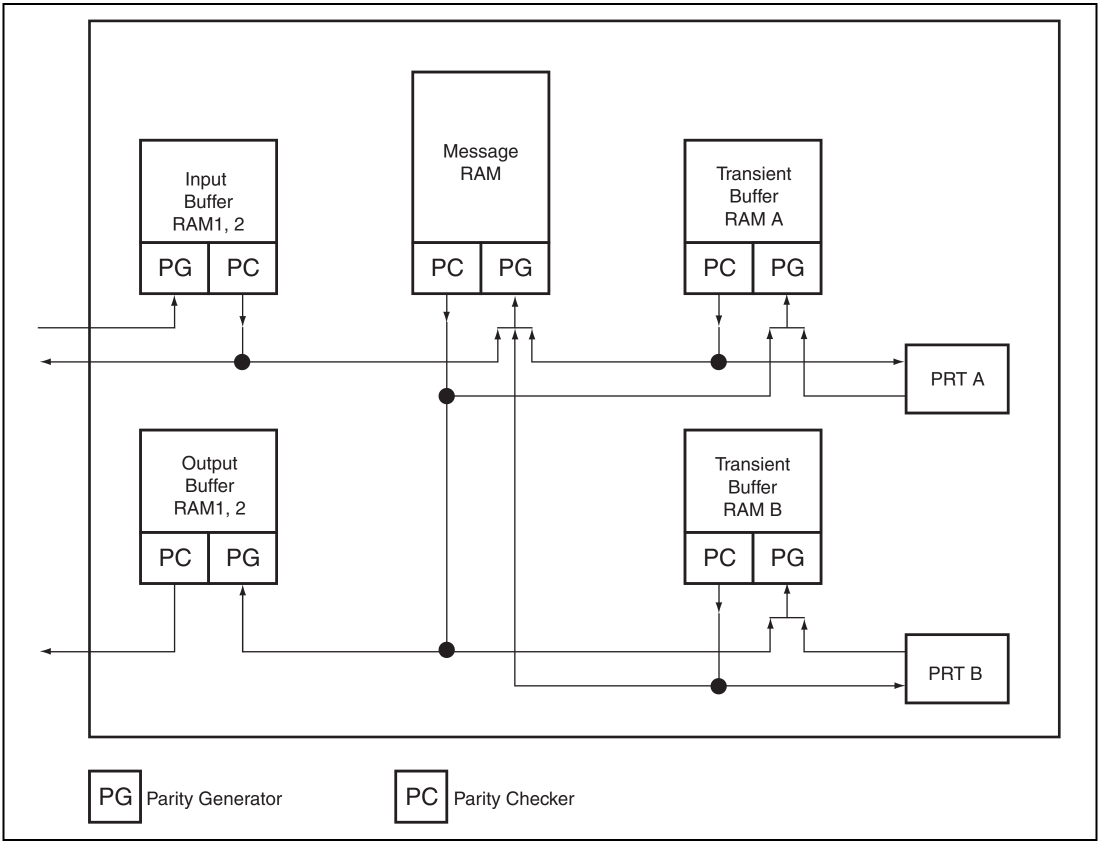


Figure 19.16 Parity Generation and Check

Note: Parity generator & checker are not part of the RAM blocks, but of the RAM access logic which is part of the FlexRay core.

When a parity error has been detected the following actions will be performed:

In all cases

- The respective parity error flag in register FRMHDS is set
- The parity error flag FREIR.PERR is set and, if enabled, a module interrupt to the Host will be generated.

Additionally in specific cases

(1) Parity error during data transfer from Input Buffer RAM 1, 2 to Message RAM

(a) Transfer of header and/or data section

- FRMHDS.PIBF bit is set
- FRMHDS.FMBD bit is set to indicate that FRMHDS.FMB[6:0] points to a faulty message buffer
- FRMHDS.FMB[6:0] indicates the number of the faulty message buffer
- Transmit buffer: Transmission request for the respective message buffer is not set

(b) Transfer of data section only

Parity error when reading header section of respective message buffer from Message RAM.

- FRMHDS.PMR bit is set
- FRMHDS.FMBD bit is set to indicate that FRMHDS.FMB[6:0] points to a faulty message buffer
- FRMHDS.FMB[6:0] indicates the number of the faulty message buffer
- The data section of the respective message buffer is not updated
- Transmit buffer: Transmission request for the respective message buffer is not set

(2) Parity error during Host reading from Input Buffer RAM 1, 2

FRMHDS.PIBF bit is set

(3) Parity error during scan of header sections in Message RAM

— FRMHDS.PMR bit is set

— FRMHDS.FMBD bit is set to indicate that FRMHDS.FMB[6:0] points to a faulty message buffer

— FRMHDS.FMB[6:0] indicates the number of the faulty message buffer

— Ignore message buffer (message buffer is skipped)

(4) Parity error during data transfer from Message RAM to Transient Buffer RAM 1, 2

— FRMHDS.PMR bit is set

— FRMHDS.FMBD bit is set to indicate that FRMHDS.FMB[6:0] points to a faulty message buffer

— FRMHDS.FMB[6:0] indicates the number of the faulty message buffer

— Frame not transmitted, frames already in transmission are invalidated by setting the frame CRC to zero

(5) Parity error during data transfer from Transient Buffer RAM 1, 2 to Protocol Controller 1, 2

— FRMHDS.PTBF1, PTBF2 bit is set

— Frames already in transmission are invalidated by setting the frame CRC to zero

(6) Parity error during data transfer from Transient Buffer RAM 1, 2 to Message RAM

(a) Parity error when reading header section of respective message buffer from Message RAM

- FRMHDS.PMR bit is set
- FRMHDS.FMBD bit is set to indicate that FRMHDS.FMB[6:0] points to a faulty message buffer

- FRMHDS.FMB[6:0] indicates the number of the faulty message buffer
- The data section of the respective message buffer is not updated

(b) Parity error when reading Transient Buffer RAM 1, 2

- FRMHDS.PTBF1, PTBF2 bit is set
- FRMHDS.FMBD bit is set to indicate that FRMHDS.FMB[6:0] points to a faulty message buffer
- FRMHDS.FMB[6:0] indicates the number of the faulty message buffer

(7) Parity error during data transfer from Message RAM to Output Buffer RAM

— FRMHDS.PMR bit is set

— FRMHDS.FMBD bit is set to indicate that FRMHDS.FMB[6:0] points to a faulty message buffer

— FRMHDS.FMB[6:0] indicates the number of the faulty message buffer

(8) Parity error during Host reading Output Buffer RAM 1, 2

FRMHDS.POBF bit is set

(9) Parity error during data read of Transient Buffer RAM 1, 2

When a parity error occurs when the Message Handler reads a frame with network management information (PPI = '1') from the Transient Buffer RAM 1, 2 the corresponding FlexRay network management vector register FRNMV1 to FRNMV3 is not updated from that frame.

19.26 Module Interrupt

19.26.1 FlexRay0 and FlexRay1 Interrupt

In general, interrupts provide a close link to the protocol timing as they are triggered almost immediately when an error or status change is detected by the CC, a frame is received or transmitted, a configured timer interrupt is activated, or a stop watch event occurred. This enables the Host to react very quickly on specific error conditions, status changes, or timer events. On the other hand too many interrupts can cause the Host to miss deadlines required for the application. Therefore the CC supports enable/disable controls for each individual interrupt source separately.

An interrupt may be triggered when

- An error was detected
- A status flag is set to 1
- A timer reaches a preconfigured value

- A message transfer from Input Buffer to Message RAM or from Message RAM to Output Buffer has completed
- A stop watch event occurred

Tracking status and generating interrupts when a status change or an error occurs are two independent tasks. Regardless of whether an interrupt is enabled or not, the corresponding status is tracked and indicated by the CC. The Host has access to the actual status and error information by reading registers FREIR and FRSIR.

Table 19.18 Module Interrupt Flags and Interrupt Line Enable Flags

Register	Bit	Function
FREIR	PEMC	Protocol Error Mode Changed
	CNA	Command Not Valid
	SFBM	Sync Frames Below Minimum
	SFO	Sync Frame Overflow
	CCF	Clock Correction Failure
	CCL	CHI Command Locked
	PERR	Parity Error
	RFO	Receive FIFO Overrun
	EFA	Empty FIFO Access
	IIBA	Illegal Input Buffer Access
	IOBA	Illegal Output Buffer Access
	MHF	Message Handler Constraints Flag
	EDA	Error Detected on Channel A
	LTVA	Latest Transmit Violation Channel A
	TABA	Transmission Across Boundary Channel A
	EDB	Error Detected on Channel B
	LTVB	Latest Transmit Violation Channel B
	TABB	Transmission Across Boundary Channel B

Register	Bit	Function
FRSIR	WST	Wakeup Status
	CAS	Collision Avoidance Symbol
	CYCS	Cycle Start Interrupt
	TXI	Transmit Interrupt
	RXI	Receive Interrupt
	RFNE	Receive FIFO not Empty
	RFCL	Receive FIFO Critical Level
	NMVC	Network Management Vector Changed
	TI0	Timer Interrupt 0
	TI1	Timer Interrupt 1
	TIBC	Transfer Input Buffer Completed
	TOBC	Transfer Output Buffer Completed
	SWE	Stop Watch Event
	SUCS	Startup Completed Successfully
	MBSI	Message Buffer Status Interrupt
	SDS	Start of Dynamic Segment
	WUPA	Wakeup Pattern Channel A
	MTSA	MTS Received on Channel A
	WUPB	Wakeup Pattern Channel B
	MTSB	MTS Received on Channel B
FRILE	EINT0	Enable Interrupt Line 0
	EINT1	Enable Interrupt Line 1

The FlexRay0 and FlexRay1 interrupt lines are controlled by the enabled interrupts. In addition, each of the two interrupt lines can be enabled/disabled separately by programming bits FRILE.EINT0 and FRILE.EINT1.

When a data transfer between IBF/OBF and the Message RAM has completed, bit FRSIR.TIBC or FRSIR.TOBC is set.

19.26.2 FlexRay Timer 0 and 1 Interrupts

The two timer interrupts generated by interrupt timer 0 and 1 are allocated as the FlexRay timer 0 and 1 interrupts. These interrupts can be set in the FRT0C and FRT1C registers. Since disabling or enabling of the interrupts by settings in the FRSIES and FRSIER registers is not possible, settings must be made in the FlexRay timer enable register (FXRTIER).

19.27 Usage Notes

19.27.1 External Clock Input

When using the FlexRay Module, the peripheral clock frequency should be set to 40 MHz because the FlexRay clock frequency (Fϕ) is set to 80 MHz.

19.27.2 Notes on Data Transfer by A-DMAC

When the A-DMAC is used to transfer data in the receive message buffer or receive FIFO to on-chip RAM, the payload length of the FlexRay Module must be the same as the payload length of the A-DMAC.

	FlexRay Module	**A-DMAC**
Payload Length	**FRMHDC.SFDL[6:0]** **FRRDHS2.PLC[6:0]**	**ADMAFRCTR.ADLNG[1:0]** **ADMAFRFCTR.ADFLNG[1:0]**
32 bytes	16	00b
64 bytes	32	01b
128 bytes	64	10b
254 bytes	127	11b

19.28 Assignment of FlexRay Configuration Parameters

Table 19.19 FlexRay Configuration Parameters

Parameter	Bit (Field)
pKeySlotusedForStartup	FRSUCC1.TXST
pKeySlotUsedForSync	FRSUCC1.TXSY
gColdStartAttempts	FRSUCC1.CSA[4:0]
pAllowPassiveToActive	FRSUCC1.PTA[4:0]
pWakeupChannel	FRSUCC1.WUCS
pSingleSlotEnabled	FRSUCC1.TSM
pAllowHaltDueToClock	FRSUCC1.HCSE
pChannels	FRSUCC1.CCHA FRSUCC1.CCHB
pdListenTimeOut	FRSUCC2.LT[20:0]
gListenNoise	FRSUCC2.LTN[3:0]
gMaxWithoutClockCorrectionPassive	FRSUCC3.WCP[3:0]
gMaxWithoutClockCorrectionFatal	FRSUCC3.WCF[3:0]
gNetworkManagementVectorLength	FRNEMC.NML[3:0]
gdTSSTransmitter	FRPRTC1.TSST[3:0]
gdCASRxLowMax	FRPRTC1.CASM[6:0]
gdSampleClockPeriod	FRPRTC1.BRP[1:0]
pSamplesPerMicrotick	FRPRTC1.BRP[1:0]
gdWakeupSymbolRxWindow	FRPRTC1.RXW[8:0]
pWakeupPattern	FRPRTC1.RWP[5:0]
gdWakeupSymbolRxIdle	FRPRTC2.RXI[5:0]
gdWakeupSymbolRxLow	FRPRTC2.RXL[5:0]
gdWakeupSymbolTxIdle	FRPRTC2.TXI[7:0]
gdWakeupSymbolTxLow	FRPRTC2.TXL[5:0]
gPayloadLengthStatic	FRMHDC.SFDL[6:0]
pLatestTx	FRMHDC.SLT[12:0]
pMicroPerCycle	FRGTUC1.UT[19:0]
gMacroPerCycle	FRGTUC2.MPC[13:0]

Parameter	Bit (Field)
gSyncNodeMax	FRGTUC2.SNM[3:0]
pMicroInitialOffset[A]	FRGTUC3.UIOA[7:0]
pMicroInitialOffset[B]	FRGTUC3.UIOB[7:0]
pMacroInitialOffset[A]	FRGTUC3.MIOA[6:0]
pMacroInitialOffset[B]	FRGTUC3.MIOB[6:0]
gdNIT	FRGTUC4.NIT[13:0]
gOffsetCorrectionStart	FRGTUC4.OCS[13.0]
pDelayCompensation[A]	FRGTUC5.DCA[7:0]
pDelayCompensation[B]	FRGTUC5.DCB[7:0]
pClusterDriftDamping	FRGTUC5.CDD[4:0]
pDecodingCorrection	FRGTUC5.DEC[7:0]
pdAcceptedStartupRange	FRGTUC6.ASR[10:0]
pdMaxDrift	FRGTUC6.MOD[10:0]
gdStaticSlot	FRGTUC7.SSL[9:0]
gNumberOfStaticSlots	FRGTUC7.NSS[9:0]
gdMinislot	FRGTUC8.MSL[5:0]
gNumberOfMinislots	FRGTUC8.NMS[12:0]
gdActionPointOffset	FRGTUC9.APO[5:0]
gdMinislotActionPoint	FRGTUC9.MAPO[4:0]
gdDynamicSlotIdlePhase	FRGTUC9.DSI[1:0]
pOffsetCorrectionOut	FRGTUC10.MOC[13:0]
pRateCorrectionOut	FRGTUC10.MRC[10:0]
pExternOffsetCorrection	FRGTUC11.EOC[2:0]
pExternRateCorrection	FRGTUC11.ERC[2:0]

Section 20 A/D Converter (ADC)

This LSI includes a 12-bit successive approximation A/D converter, which consists of two independent units (ADC_A and ADC_B). Up to 37 channel analog inputs can be selected by software.

20.1 Features

- Resolution: 12 bits
- Input channels: 37 channels (ADC_A: 28 channels (AN0 to AN27), ADC_B: 9 channels (AN40 to AN48)
- Minimum conversion time

  1.25 μs/channel at Pϕ = 40MHz operation (conversion state = 50 states)

  Low-speed setting: 2.5 μs/channel (operation at Pϕ = 20 MHz, conversion state = 50 states)

  High-speed setting: 1.25 μs/channel (operation at Pϕ = 20 MHz, conversion state = 25 states)
- Two scan conversion modes are selectable

  Single cycle scan mode: scanning only once

  Continuous scan mode: scanning repeatedly

  Channels to be scanned can be selected as desired, and A/D conversion is in ascending order of channel number (ADC_A: AN0 → AN27, ADC_B: AN40 → AN48).
- A/D-converted value addition mode

  The same channel is A/D converted continuously 2 to 4 times. The sum of the converted values is stored in the A/D data register. AN0 to AN27 and AN40 to AN48 can be used in A/D-converted value addition mode.

  The use of the average of these results can improve the precision of A/D conversion, depending on the types of noise components that are present. This function, however, cannot guarantee an improvement in A/D conversion accuracy.
- Thirty seven 14-bit A/D data registers
- Sample and hold function

  Each A/D converter module (ADC_A and ADC_B) includes a sample & hold circuit.
- Two types of scan conversion start

  ADC_A: Software trigger (ADST bit in ADCSR0), external trigger (ADTRG_A) or ATU-III timer trigger (timer G4) can be selected.

  ADC_B: Software trigger (ADST bit in ADCSR1), external trigger (ADTRG_B) or ATU-III timer trigger (timer G5) can be selected.

- Interrupt-triggered conversion

  Independently from the scan conversion, it is possible to preferentially process channels requested by an ATU-III timer trigger or software trigger for A/D conversion. AN0 to AN15 and AN40 to AN47 support this function.

  If an interrupt conversion and a scan conversion is requested simultaneously, the scan conversion is suspended and the A/D conversion is preferentially executed on the channel in which the interrupt conversion is requested. On completion of the interrupt conversion, the scan conversion is resumed from the A/D conversion on the interrupted channel.
- Supporting scan conversion end interrupt (ADI), interrupt conversion end interrupt (ADID), and DMA transfer function

  On completion of scanning for scan conversion, a scan conversion end interrupt request (ADI) can be generated or the DMAC can be started. On completion of interrupt conversion on channels AN0 to AN15 or AN40 to AN47, an interrupt conversion end interrupt request (ADID0 to ADID15 and ADID40 to ADID47) can be generated, or A-DMAC (ADID0) or DMAC (ADID40 to ADID47) can be started up.
- Offset calibration function

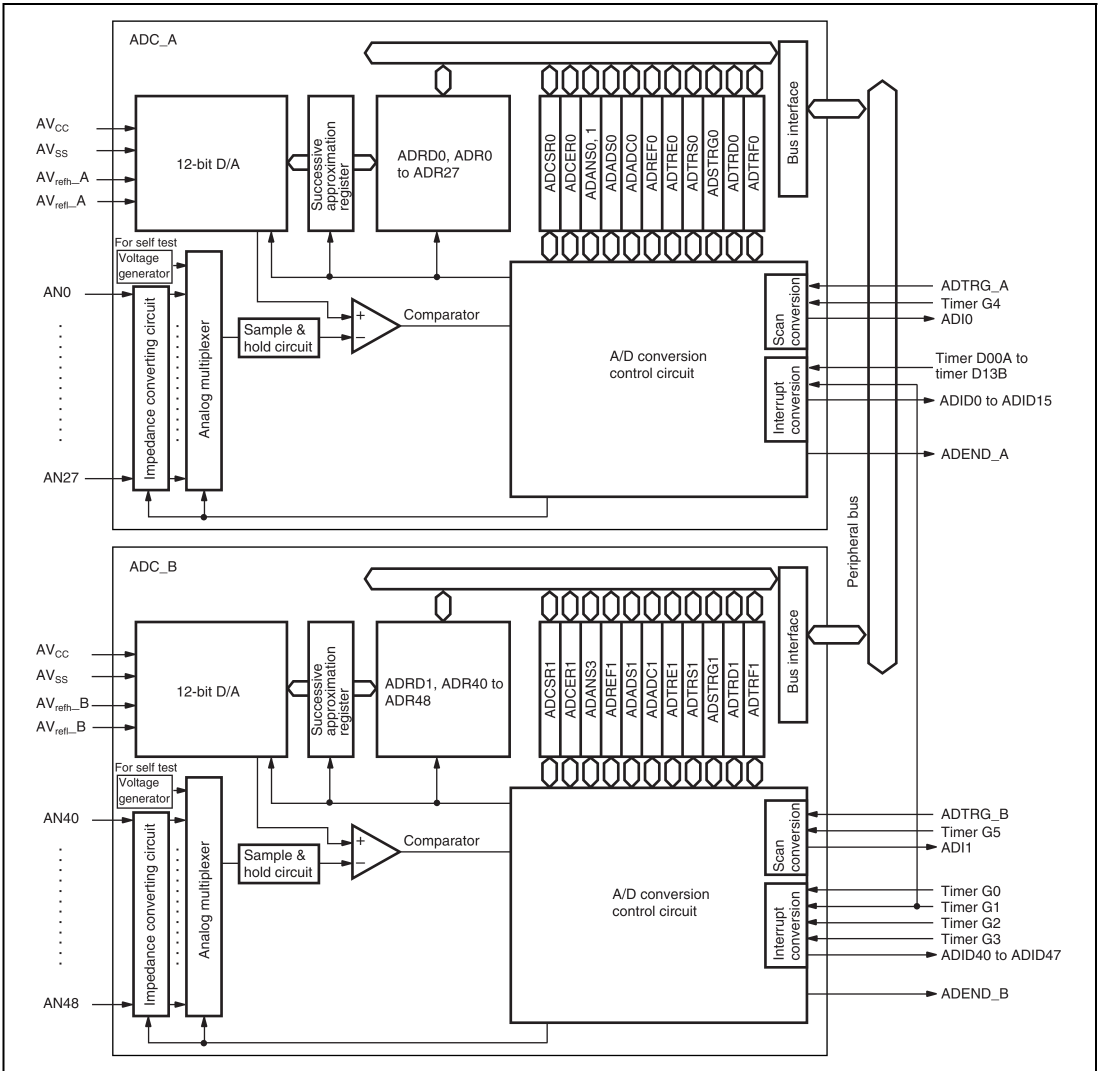
  The offset error due to temperature variation is corrected to achieve a high accuracy A/D conversion.
- Self-test function of A/D converter

  The self-test function of A/D converter performs the A/D conversion of internally generated voltage values (AVref (AVrefh_A, AVrefh_B) × 0, AVref × 1/4, AVref × 1/2, AVref × 3/4, and AVref × 1), and returns A/D converted values and information on converted voltages to A/D data registers Diag0 and Diag1. After that, whether or not the A/D converted values (the contents of the above A/D data registers) are in the normal range is checked by software to detect errors in the A/D converter.
- Programmable analog input voltage range

  Analog input voltage range is programmable via the AVrefh_A pin.
- ADEND output

  When channels AN0 and AN40 are used for scan conversion, the conversion timing signals are output via pins ADEND_A and ADEND_B.

Figure 20.1 shows a block diagram of the A/D converter.



[Legend]

ADRD:	A/D data register diag
ADR:	A/D data register
ADCSR:	A/D control register
ADREF:	A/D conversion status register
ADTRE:	A/D interrupt trigger enable register
ADTRF:	A/D interrupt trigger conversion end flag register
ADTRS:	A/D interrupt trigger source select register
ADSTRG:	A/D interrupt software trigger register
ADTRD:	A/D interrupt trigger conversion interrupt enable register
ADADS:	A/D converted value addition mode select register
ADADC:	A/D converted value addition count select register
ADANS:	A/D channel select register
ADCER:	A/D control extended register

Figure 20.1 Block Diagram of A/D Converter

20.2 Input/Output Pins

Table 20.1 shows the pin configuration of the A/D converter.

For LSI reliability assurance, when using the A/D converter, ensure that the following is satisfied:

$AV_{CC} = 5.0\ V \pm 0.5\ V,\ AV_{SS} = V_{SS}$

When the A/D converter is not used, neither the AV_{CC} nor AV_{SS} pins must be left open. The voltage applied to the analog input pins must be within the following range.

AVrefl_A ≤ ANn (n = 0 to 27, 40 to 48) ≤ AVrefh_A

AVrefl_B ≤ ANn (n = 0 to 27, 40 to 48) ≤ AVrefh_B

Table 20.1 Pin Configuration

Pin Name	Symbol	I/O	Function
AV_{CC}	AV_{CC}	Input	Analog power supply
AV_{SS}	AV_{SS}	Input	Analog ground
AVREFL_A	AVrefl_A	Input	Input pin for analog reference voltage for ADC_A (AVrefl_A < AVrefh_A)
AVREFH_A	AVrefh_A	Input	Input pin for analog reference voltage for ADC_A (AVrefl_A < AVrefh_A)
AVREFL_B	AVrefl_B	Input	Input pin for analog reference voltage for ADC_B (AVrefl_B < AVrefh_B)
AVREFH_B	AVrefh_B	Input	Input pin for analog reference voltage for ADC_B (AVrefl_B < AVrefh_B)
AN_A0	AN0	Input	Analog input channel 0
AN_A1	AN1	Input	Analog input channel 1
AN_A2	AN2	Input	Analog input channel 2
AN_A3	AN3	Input	Analog input channel 3
AN_A4	AN4	Input	Analog input channel 4
AN_A5	AN5	Input	Analog input channel 5
AN_A6	AN6	Input	Analog input channel 6
AN_A7	AN7	Input	Analog input channel 7
AN_A8	AN8	Input	Analog input channel 8
AN_A9	AN9	Input	Analog input channel 9
AN_A10	AN10	Input	Analog input channel 10
AN_A11	AN11	Input	Analog input channel 11
AN_A12	AN12	Input	Analog input channel 12
AN_A13	AN13	Input	Analog input channel 13
AN_A14	AN14	Input	Analog input channel 14
AN_A15	AN15	Input	Analog input channel 15
AN_A16	AN16	Input	Analog input channel 16
AN_A17	AN17	Input	Analog input channel 17
AN_A18	AN18	Input	Analog input channel 18
AN_A19	AN19	Input	Analog input channel 19

Pin Name	Symbol	I/O	Function
AN_A20	AN20	Input	Analog input channel 20
AN_A21	AN21	Input	Analog input channel 21
AN_A22	AN22	Input	Analog input channel 22
AN_A23	AN23	Input	Analog input channel 23
AN_A24	AN24	Input	Analog input channel 24
AN_A25	AN25	Input	Analog input channel 25
AN_A26	AN26	Input	Analog input channel 26
AN_A27	AN27	Input	Analog input channel 27
AN_B40	AN40	Input	Analog input channel 40
AN_B41	AN41	Input	Analog input channel 41
AN_B42	AN42	Input	Analog input channel 42
AN_B43	AN43	Input	Analog input channel 43
AN_B44	AN44	Input	Analog input channel 44
AN_B45	AN45	Input	Analog input channel 45
AN_B46	AN46	Input	Analog input channel 46
AN_B47	AN47	Input	Analog input channel 47
AN_B48	AN48	Input	Analog input channel 48
ADTRG_A	ADTRG_A	Input	Input pin for scan conversion trigger of ADC_A
ADTRG_B	ADTRG_B	Input	Input pin for scan conversion trigger of ADC_B
ADEND_A	ADEND_A	Output	Output pin for monitoring AN0 conversion timing of ADC_A
ADEND_B	ADEND_B	Output	Output pin for monitoring AN40 conversion timing of ADC_B

20.3 Functions Assigned to Each Channel

Table 20.2 lists the functions assigned to each channel.

Table 20.2 List of Functions Assigned to Each Channel

Unit of A/D Converter	A/D Conversion Time/Channel	A/D Conversion Channel		Scan conversion								A/D data register (Register storing A/D-converted value)	A/D converted value addition mode	A/D conversion timing monitor pin (ADEND_A and ADEND_B)	Interrupt conversion									
				Trigger source			Self Test/Channel	Scan conversion mode	End interrupt/DMA transfer request signal	DMA transfer function (DMAC)	End flag (upon completion of a scan)				Trigger source			End interrupt DMA activation request	DMA transfer function		Enable (channel selection)	Trigger source selection	End flag	End interrupt enable
				Source 1	Source 2	Source 3									Source 1	Source 2	Source 3							
		Self Test (Internally Generated Voltage)/ Pin Name	Sym-bol	Software	Pin	ATU-III	ADCER0 and ADCER1/ ADANS0 and ADANS1						ADADS0 and ADADS0 to 5		ATU-III		Software (ADSTRG0 and ADSTRG1)		DMAC	A-DMAC	ADTRE0 and ADTRE1	ADTRS0 and ADTRS1	ADTRF0 and ADTRF1	ADTRD0 and ADTRD1
A/D converter_A (ADC_A)	2.5 µs/1.25 µs	Self test 0	DIAG0	ADST (ADCSR0)	External trigger_A (ADTRG_A)	Timer G4	DIAGM (ADCER0)	Single cycle scanning or continuous scanning	ADI0	Yes	ADF (ADREF0)	ADRD0	No	No	No	No	No	No	No	No	No	No	No	No
		Analog input pin 0	AN0				ANS0					ADR0	ADS0	Yes	Timer D00A	Timer G1	ADSTRG0	ADID0	No	Yes	ADTRGE0	ADTRS0	ADTF0	ADIDE0
		Analog input pin 1	AN1				ANS1					ADR1	ADS1	No	Timer D00B	No	ADSTRG1	ADID1	No	No	ADTRGE1	ADTRS1	ADTF1	ADIDE1
		Analog input pin 2	AN2				ANS2					ADR2	ADS2	No	Timer D01A	No	ADSTRG2	ADID2	No	No	ADTRGE2	ADTRS2	ADTF2	ADIDE2
		Analog input pin 3	AN3				ANS3					ADR3	ADS3	No	Timer D01B	No	ADSTRG3	ADID3	No	No	ADTRGE3	ADTRS3	ADTF3	ADIDE3
		Analog input pin 4	AN4				ANS4					ADR4	ADS4	No	Timer D02A	No	ADSTRG4	ADID4	No	No	ADTRGE4	ADTRS4	ADTF4	ADIDE4
		Analog input pin 5	AN5				ANS5					ADR5	ADS5	No	Timer D02B	No	ADSTRG5	ADID5	No	No	ADTRGE5	ADTRS5	ADTF5	ADIDE5
		Analog input pin 6	AN6				ANS6					ADR6	ADS6	No	Timer D03A	No	ADSTRG6	ADID6	No	No	ADTRGE6	ADTRS6	ADTF6	ADIDE6
		Analog input pin 7	AN7				ANS7					ADR7	ADS7	No	Timer D03B	No	ADSTRG7	ADID7	No	No	ADTRGE7	ADTRS7	ADTF7	ADIDE7
		Analog input pin 8	AN8				ANS8					ADR8	ADS8	No	Timer D10A	No	ADSTRG8	ADID8	No	No	ADTRGE8	ADTRS8	ADTF8	ADIDE8
		Analog input pin 9	AN9				ANS9					ADR9	ADS9	No	Timer D10B	No	ADSTRG9	ADID9	No	No	ADTRGE9	ADTRS9	ADTF9	ADIDE9
		Analog input pin 10	AN10				ANS10					ADR10	ADS10	No	Timer D11A	No	ADSTRG10	ADID10	No	No	ADTRGE10	ADTRS10	ADTF10	ADIDE10
		Analog input pin 11	AN11				ANS11					ADR11	ADS11	No	Timer D11B	No	ADSTRG11	ADID11	No	No	ADTRGE11	ADTRS11	ADTF11	ADIDE11
		Analog input pin 12	AN12				ANS12					ADR12	ADS12	No	Timer D12A	No	ADSTRG12	ADID12	No	No	ADTRGE12	ADTRS12	ADTF12	ADIDE12
		Analog input pin 13	AN13				ANS13					ADR13	ADS13	No	Timer D12B	No	ADSTRG13	ADID13	No	No	ADTRGE13	ADTRS13	ADTF13	ADIDE13
		Analog input pin 14	AN14				ANS14					ADR14	ADS14	No	Timer D13A	No	ADSTRG14	ADID14	No	No	ADTRGE14	ADTRS14	ADTF14	ADIDE14
		Analog input pin 15	AN15				ANS15					ADR15	ADS15	No	Timer D13B	No	ADSTRG15	ADID15	No	No	ADTRGE15	ADTRS15	ADTF15	ADIDE15
		Analog input pin 16	AN16				ANS16					ADR16	ADS16	No	No	No	No	No	No	No	No	No	No	No
		Analog input pin 17	AN17				ANS17					ADR17	ADS17	No	No	No	No	No	No	No	No	No	No	No
		Analog input pin 18	AN18				ANS18					ADR18	ADS18	No	No	No	No	No	No	No	No	No	No	No
		Analog input pin 19	AN19				ANS19					ADR19	ADS19	No	No	No	No	No	No	No	No	No	No	No
		Analog input pin 20	AN20				ANS20					ADR20	ADS20	No	No	No	No	No	No	No	No	No	No	No
		Analog input pin 21	AN21				ANS21					ADR21	ADS21	No	No	No	No	No	No	No	No	No	No	No
		Analog input pin 22	AN22				ANS22					ADR22	ADS22	No	No	No	No	No	No	No	No	No	No	No
		Analog input pin 23	AN23				ANS23					ADR23	ADS23	No	No	No	No	No	No	No	No	No	No	No
		Analog input pin 24	AN24				ANS24					ADR24	ADS24	No	No	No	No	No	No	No	No	No	No	No
		Analog input pin 25	AN25				ANS25					ADR25	ADS25	No	No	No	No	No	No	No	No	No	No	No
		Analog input pin 26	AN26				ANS26					ADR26	ADS26	No	No	No	No	No	No	No	No	No	No	No
		Analog input pin 27	AN27				ANS27					ADR27	ADS27	No	No	No	No	No	No	No	No	No	No	No
A/D converter_B (ADC_B)	2.5 µs/1.25 µs	Self test 1	DIAG1	ADST (ADCSR1)	External trigger_B (ADTRG_B)	Timer G5	DIAGM (ADCER1)	Single scan or continuous scan	ADI1	Yes	ADF (ADREF1)	ADRD1	ADS28	No	No	No	No	No	No	No	No	No	No	No
		Analog input pin 40	AN40				ANS40					ADR40	ADS40	Yes	Timer G0	No	ADSTRG40	ADID40	Yes	No	ADTRGE40	ADTRS40	ADTF40	ADIDE40
		Analog input pin 41	AN41				ANS41					ADR41	ADS41	No	Timer G1	No	ADSTRG41	ADID41	Yes	No	ADTRGE41	ADTRS41	ADTF41	ADIDE41
		Analog input pin 42	AN42				ANS42					ADR42	ADS42	No	Timer G2	No	ADSTRG42	ADID42	Yes	No	ADTRGE42	ADTRS42	ADTF42	ADIDE42
		Analog input pin 43	AN43				ANS43					ADR43	ADS43	No	Timer G3	No	ADSTRG43	ADID43	Yes	No	ADTRGE43	ADTRS43	ADTF43	ADIDE43
		Analog input pin 44	AN44				ANS44					ADR44	ADS44	No	Timer G0	No	ADSTRG44	ADID44	Yes	No	ADTRGE44	ADTRS44	ADTF44	ADIDE44
		Analog input pin 45	AN45				ANS45					ADR45	ADS45	No	Timer G1	No	ADSTRG45	ADID45	Yes	No	ADTRGE45	ADTRS45	ADTF45	ADIDE45
		Analog input pin 46	AN46				ANS46					ADR46	ADS46	No	Timer G2	No	ADSTRG46	ADID46	Yes	No	ADTRGE46	ADTRS46	ADTF46	ADIDE46
		Analog input pin 47	AN47				ANS47					ADR47	ADS47	No	Timer G3	No	ADSTRG47	ADID47	Yes	No	ADTRGE47	ADTRS47	ADTF47	ADIDE47
		Analog input pin 48	AN48				ANS48					ADR48	ADS48	No	No	No	No	No	No	No	No	No	No	No

20.4 Register Descriptions

The A/D converter has the following registers.

Table 20.3 Register Configuration

Register Name	Symbol	R/W	Initial Value	Address	Unit	Access Size*[1]
A/D data register Diag0	ADRD0	R	H'0000	H'FFFFE83E	ADC_A	16
A/D data register 0	ADR0	R	H'0000	H'FFFFE840	ADC_A	16
A/D data register 1	ADR1	R	H'0000	H'FFFFE842	ADC_A	16
A/D data register 2	ADR2	R	H'0000	H'FFFFE844	ADC_A	16
A/D data register 3	ADR3	R	H'0000	H'FFFFE846	ADC_A	16
A/D data register 4	ADR4	R	H'0000	H'FFFFE848	ADC_A	16
A/D data register 5	ADR5	R	H'0000	H'FFFFE84A	ADC_A	16
A/D data register 6	ADR6	R	H'0000	H'FFFFE84C	ADC_A	16
A/D data register 7	ADR7	R	H'0000	H'FFFFE84E	ADC_A	16
A/D data register 8	ADR8	R	H'0000	H'FFFFE850	ADC_A	16
A/D data register 9	ADR9	R	H'0000	H'FFFFE852	ADC_A	16
A/D data register 10	ADR10	R	H'0000	H'FFFFE854	ADC_A	16
A/D data register 11	ADR11	R	H'0000	H'FFFFE856	ADC_A	16
A/D data register 12	ADR12	R	H'0000	H'FFFFE858	ADC_A	16
A/D data register 13	ADR13	R	H'0000	H'FFFFE85A	ADC_A	16
A/D data register 14	ADR14	R	H'0000	H'FFFFE85C	ADC_A	16
A/D data register 15	ADR15	R	H'0000	H'FFFFE85E	ADC_A	16
A/D data register 16	ADR16	R	H'0000	H'FFFFE860	ADC_A	16
A/D data register 17	ADR17	R	H'0000	H'FFFFE862	ADC_A	16
A/D data register 18	ADR18	R	H'0000	H'FFFFE864	ADC_A	16
A/D data register 19	ADR19	R	H'0000	H'FFFFE866	ADC_A	16
A/D data register 20	ADR20	R	H'0000	H'FFFFE868	ADC_A	16
A/D data register 21	ADR21	R	H'0000	H'FFFFE86A	ADC_A	16
A/D data register 22	ADR22	R	H'0000	H'FFFFE86C	ADC_A	16
A/D data register 23	ADR23	R	H'0000	H'FFFFE86E	ADC_A	16
A/D data register 24	ADR24	R	H'0000	H'FFFFE870	ADC_A	16

Register Name	Symbol	R/W	Initial Value	Address	Unit	Access Size*[1]
A/D data register 25	ADR25	R	H'0000	H'FFFFE872	ADC_A	16
A/D data register 26	ADR26	R	H'0000	H'FFFFE874	ADC_A	16
A/D data register 27	ADR27	R	H'0000	H'FFFFE876	ADC_A	16
A/D data register Diag1	ADRD1	R	H'0000	H'FFFFEC3E	ADC_B	16
A/D data register 40	ADR40	R	H'0000	H'FFFFEC40	ADC_B	16
A/D data register 41	ADR41	R	H'0000	H'FFFFEC42	ADC_B	16
A/D data register 42	ADR42	R	H'0000	H'FFFFEC44	ADC_B	16
A/D data register 43	ADR43	R	H'0000	H'FFFFEC46	ADC_B	16
A/D data register 44	ADR44	R	H'0000	H'FFFFEC48	ADC_B	16
A/D data register 45	ADR45	R	H'0000	H'FFFFEC4A	ADC_B	16
A/D data register 46	ADR46	R	H'0000	H'FFFFEC4C	ADC_B	16
A/D data register 47	ADR47	R	H'0000	H'FFFFEC4E	ADC_B	16
A/D data register 48	ADR48	R	H'0000	H'FFFFEC50	ADC_B	16
A/D control register 0	ADCSR0	R/W	H'00	H'FFFFE800	ADC_A	8
A/D control register 1	ADCSR1	R/W	H'00	H'FFFFEC00	ADC_B	8
A/D conversion status register 0	ADREF0	R/(W)*[2]	H'00	H'FFFFE802	ADC_A	8
A/D conversion status register 1	ADREF1	R/(W)*[2]	H'00	H'FFFFEC02	ADC_B	8
A/D interrupt trigger enable register 0	ADTRE0	R/W	H'0000	H'FFFFE804	ADC_A	8, 16
A/D interrupt trigger enable register 1	ADTRE1	R/W	H'00	H'FFFFEC10	ADC_B	8
A/D interrupt trigger conversion end flag register 0	ADTRF0	R/(W)*[2]	H'0000	H'FFFFE806	ADC_A	8, 16
A/D interrupt trigger conversion end flag register 1	ADTRF1	R/(W)*[2]	H'00	H'FFFFEC12	ADC_B	8
A/D interrupt trigger source select register 0	ADTRS0	R/W	H'0000	H'FFFFE808	ADC_A	8, 16
A/D interrupt trigger source select register 1	ADTRS1	R/W	H'00	H'FFFFEC14	ADC_B	8
A/D interrupt software trigger register 0	ADSTRG0	W	H'0000	H'FFFFE80A	ADC_A	8, 16
A/D interrupt software trigger register 1	ADSTRG1	W	H'00	H'FFFFEC16	ADC_B	8
A/D interrupt trigger conversion end interrupt enable register 0	ADTRD0	R/W	H'0000	H'FFFFE80C	ADC_A	8, 16

Register Name	Symbol	R/W	Initial Value	Address	Unit	Access Size*1
A/D interrupt trigger conversion end interrupt enable register 1	ADTRD1	R/W	H'00	H'FFFFEC18	ADC_B	8
A/D-converted value addition mode select register 0	ADADS0	R/W	H'00	H'FFFFE81C	ADC_A	8
A/D-converted value addition mode select register 1	ADADS1	R/W	H'00	H'FFFFEC1C	ADC_B	8
A/D-converted value addition mode select register 2	ADADS2	R/W	H'00	H'FFFFE924	ADC_A	8
A/D-converted value addition mode select register 3	ADADS3	R/W	H'00	H'FFFFE926	ADC_A	8
A/D-converted value addition mode select register 4	ADADS4	R/W	H'00	H'FFFFE928	ADC_A	8
A/D-converted value addition mode select register 5	ADADS5	R/W	H'00	H'FFFFED24	ADC_B	8
A/D-converted value addition count selection register 0	ADADC0	R/W	H'00	H'FFFFE81E	ADC_A	8
A/D-converted value addition count selection register 1	ADADC1	R/W	H'00	H'FFFFEC1E	ADC_B	8
A/D channel select register 0	ADANS0	R/W	H'0000	H'FFFFE820	ADC_A	8, 16, 32
A/D channel select register 1	ADANS1	R/W	H'0000	H'FFFFE822	ADC_A	8, 16, 32
A/D channel select register 3	ADANS3	R/W	H'0000	H'FFFFEC20	ADC_B	8, 16
A/D control extended register 0	ADCER0	R/W	H'0000	H'FFFFE830	ADC_A	8, 16
A/D control extended register 1	ADCER1	R/W	H'0000	H'FFFFEC30	ADC_B	8, 16

Notes: 1. 16-bit access can be made only on word boundaries, while 32-bit access can be made only on longword boundaries.

2. Only 0 can be written to clear the flag.

20.4.1 A/D Data Registers 0 to 27, 40 to 48, Diag0, and Diag1 (ADR0 to ADR27, ADR40 to ADR48, ADRD0, and ADRD1)

ADR0 to ADR27 and ADR40 to ADR48 are 16-bit read-only registers which store the A/D converted results of channels AN0 to AN27 and AN40 to AN48. ADRD0 and ADRD1 are 16-bit read-only registers which store the A/D converted results of ADC_A and ADC_B for self-test. ADR0 to ADR27, ADR40 to ADR48, ADRD0, and ADRD1 are initialized to H'0000 by a power-on reset or a transition to the hardware standby mode.

ADR0 to ADR27, ADR40 to ADR48, ADRD0 and ADRD1 use different formats depending on the setting values of the A/D data register format selection (ADRFMT) bit and A/D-converted value addition mode select bits (ADS0 to ADS27 and ADS40 to ADS48). Note that A/D-converted-value addition mode can only be set for all of A/D data registers, ADR0 to ADR27 and ADR40 to ADR48. ADRD0 and ADRD1 also include the self-test status bits.

(1) ADR0 to ADR27 and ADR40 to ADR48

The ADRFMT bit can be used to set either right- or left-alignment. For this operation, the AD11 to AD0 bits indicate a 12-bit A/D converted value. Other bits are reserved. These reserved bits are always read as 0. The write value should always be 0.

In A/D-converted value addition mode, any setting in the ADRFMT bit becomes invalid. In this case, the AD13 to AD0 bits indicate a value which is obtained by adding all converted values in A/D-converted value addition mode. Other bits are reserved. These reserved bits are always read as 0. The write value should always be 0.

The following minimum and maximum values apply to channels on which A/D-converted value addition mode is selected:

- First time: H'0000 ≤ ADRn (n = 0 to 27, 40 to 48) ≤ H'3FFC
- Second time: H'0000 ≤ ADRn (n = 0 to 27, 40 to 48) ≤ H'7FF8
- Third time: H'0000 ≤ ADRn (n = 0 to 27, 40 to 48) ≤ H'BFF4
- Fourth time: H'0000 ≤ ADRn (n = 0 to 27, 40 to 48) ≤ H'FFF0

- When left-alignment is selected

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AD11	AD10	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

- When right-alignment is selected

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	AD11	AD10	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

- When A/D-converted value addition mode is selected

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AD13	AD12	AD11	AD10	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

(2) ADRD0 and ADRD1

The ADRFMT bit can be used to specify either right or left-alignment. For this operation, bits AD11 to AD0 indicate a 12-bit A/D converted value. ADRD0 and ADRD1 include self-test status (DIAGST and DIAGST2) bits. Other bits are reserved. These reserved bits are always read as 0. The write value should always be 0. ADR0 and ADR1 cannot be used in A/D-converted value addition mode.

- When left-alignment is specified

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AD11	AD10	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	-	DIAG ST2	DIAGST[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 4	AD11 to AD0	All 0	R	12-bit A/D-Converted Value
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	DIAGST2	0	R	Self-Test Status 2 For details on self-testing, see the description of DIAGST bit.
1, 0	DIAGST[1:0]	00	R	Self-Test Status These bits indicate the conversion voltage at which a self-test is performed. For details on self-test, see section 20.4.3, A/D Control Extended Registers 0 and 1 (ADCER0 and ADCER1). Read this bit and the DIAGST2 bit at the same time to check the state of self-testing. • When 0 has been read-out from the DIAGST2 bit 00: No self-test has ever been performed since power-on. 01: A self-test has been performed at a voltage AVref × 0. 10: A self-test has been performed at a voltage AVref × 1/2. 11: A self-test has been performed at a voltage AVref × 1.

Bit	Bit Name	Initial Value	R/W	Description
1, 0	DIAGST[1:0]	00	R	• When 1 has been read-out from the DIAGST2 bit 00: Reserved. 01: A self-test has been performed at a voltage AVref × 1/4. 10: A self-test has been performed at a voltage AVref × 3/4. 11: Reserved.

- When right-alignment is specified

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DIAGST[1:0]		DIAG ST2	-	AD11	AD10	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15, 14	DIAGST[1:0]	00	R	Self-test Status These bits indicate the conversion voltage at which a self-test is performed. For details on self-test, see section 20.4.3, A/D Control Extended Registers 0 and 1 (ADCER0 and ADCER1). Read these bits and the DIAGST2 bit at the same time to check the state of self-testing. • When 0 has been read-out from the DIAGST2 bit 00: No self-test has ever been performed since power-on. 01: A self-test has been performed at a voltage AVref × 0. 10: A self-test has been performed at a voltage AVref × 1/2. 11: A self-test has been performed at a voltage AVref × 1. • When 1 has been read-out from the DIAGST2 bit 00: Reserved. 01: A self-test has been performed at a voltage AVref × 1/4. 10: A self-test has been performed at a voltage AVref × 3/4. 11: Reserved.

Bit	Bit Name	Initial Value	R/W	Description
13	DIAGST2	0	R	Self-Test Status 2 For details on self-testing, see the description of the DIAGST bits.
12	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
11 to 0	AD11 to AD0	All 0	R	12-bit A/D-Converted Value

20.4.2 A/D Control Registers 0 and 1 (ADCSR0 and ADCSR1)

ADCSR0 and ADCSR1 are 8-bit readable/writable registers used to set scan conversion mode. ADCSR0 and ADCSR1 are initialized to H'00 by a power-on reset or in hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	ADST	ADCS	-	ADIE	-	-	TRGE	EXTRG
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	ADST	0	R/W	Scan Conversion Start Starts or stops scan conversion. When the ADST bit is set from 0 to 1, the A/D converter detects an ADST rising edge and then starts a scan conversion process. When the ADST bit is cleared from 1 to 0, the A/D converter detects an ADST falling edge and then stops a scan conversion process. The ADST bit does not affect interrupt conversion. To check whether a scan conversion is being performed, read the ADSCACT bit in ADREF. 0: Stops a scan conversion process. 1: Starts a scan conversion process.
6	ADCS	0	R/W	Scan Conversion Mode Select Selects scan conversion mode. To prevent incorrect operation, the ADSCACT bit in ADREF must be cleared to 0 while the ADCS value is changed. Single-cycle scan mode performs scanning once and, upon its completion, stops the scan conversion process. Continuous scan mode repeats the scanning process indefinitely. The scan conversion can be stopped by writing 0 to the ADCS bit when the bit is set to 1. Channels are A/D-converted with the lowest channel first (ADC_A: AN0 → AN27, ADC_B: AN40 → AN48). In continuous scan mode, the conversion process returns to the first channel when all the selected channels have been converted. 0: Single-cycle scan mode 1: Continuous scan mode

Bit	Bit Name	Initial Value	R/W	Description
5	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
4	ADIE	0	R/W	Interrupt Enable Enables or disables generation of the A/D scan conversion end interrupt (ADI). To prevent incorrect operation, the ADSCACT bit in ADREF must be cleared to 0 while the ADIE value is changed. When the ADF bit in ADREF is set to 1 upon completion of each scan in the scan conversion process, an ADI interrupt is generated if the ADIE bit is set to 1. The ADI interrupt can be cleared by clearing the ADF bit to 0 or clearing the ADIE bit to 0. 0: Disables ADI interrupt generation upon scanning completion. 1: Enables ADI interrupt generation upon scanning completion.
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	TRGE	0	R/W	Trigger Enable Enables or disables scan conversion to be started by an external trigger (ADTGR_A, ADTGR_B) or ATU-III timer trigger (ADC_A: timer G4, ADC_B: timer G5). 0: Disables scan conversion to be started by an external trigger or ATU-III timer trigger. 1: Enables scan conversion to be started by an external trigger or ATU-III timer trigger.
0	EXTRG	0	R/W	Trigger Select Selects a trigger source for scan conversion. Either the external trigger (ADTGR_A, ADTGR_B) or the ATU-III timer trigger (ADC_A: timer G4, ADC_B: timer G5) is selected. 0: Scan conversion is started by an ATU-III timer trigger. 1: Scan conversion is started by an external trigger.

Notes: 1. Starting ADC_A and ADC_B scan conversion simultaneously

ADC_A and ADC_B scan conversion can be started simultaneously by writing 1 to the TRGE bit and 0 to the EXTRG bit in both ADCSR0 and ADCSR1, and by inputting the timer G4 trigger and the timer G5 trigger from the ATU-III simultaneously. For details on the timer G settings, see section 13, Advanced Timer Unit III (ATU-III).

2. Starting ADC_A and ADC_B scan conversion with different timing from each other

ADC_A and ADC_B scan conversion can be started with different timing from each other by writing 1 to the TRGE bit and 0 to the EXTRG bit in ADCSR0 and ADCSR1, and by inputting the timer G4 trigger and the timer G5 trigger from the ATU-III with different timing from each other. For details on the timer G settings, see section 13, Advanced Timer Unit III (ATU-III).

3. Staring an interrupt conversion and a scan conversion simultaneously

If 1 is written to the TRGE bit and 0 is written to the EXTRG bit in ADCSR1, and 1 is written to the ADTRGE40 bit and the ADTRGE44 bit in the A/D interrupt trigger enable register (ADTRE1), and then the timer G5 trigger and the timer G0 trigger are simultaneously input from the ATU-III, the ADC_B executes the following operations in the indicated sequence: AN40 interrupt conversion → AN44 interrupt conversion → scan conversion. To execute a scan conversion only, clear both the ADTRGE40 and ADTRGE44 bits to 0. Either AN40 or AN44 can also execute a single-channel interrupt conversion. Similar operations can also be accomplished through combinations of an ADC_A scan conversion and an AN0 interrupt conversion by using the timer G4 trigger and the timer G1 trigger from the ATU-III.

4. Starting a scan conversion using an external trigger

If 1 is written to both the TRGE and EXTRG bits when high level signals are input to the external trigger pins (ADTRG_A and ADTRG_B), and then if a low-level pulse is input to either ADTRG_A or ADTRG_B, either ADC_A or ADC_B detects a pulse falling edge and starts the scan conversion process. In this case, the low pulse width must be 1.5Pϕ clock or more.

5. Independent of the ADST bit, external triggers and ATU-III timer triggers, startup of a scan conversion is enabled when the ADSCACT bit in the A/D conversion status register (ADREF) is cleared to 0. The startup source for a scan conversion is not retained.

6. Note on the cycle time of starting scan conversion or interrupt conversion by the timer trigger from the ATU-III:
Set the ATU-III timer trigger cycle such that it is longer than the scan conversion time and the interrupt conversion time. (Example for scan conversion: in the case of conversion on a single channel, 56 states when bit CKS = 0 and 28 states when CKS = 1. Example for interrupt conversion: in the case of starting conversion on a single channel using one trigger source, 50 states when bit CKS = 0 and 25 states when CKS = 1). For details on the timer trigger cycle setting, see section 13, Advanced Timer Unit III (ATU-III).

20.4.3 A/D Control Extended Registers 0 and 1 (ADCER0 and ADCER1)

ADCER0 and ADCER1 are 16-bit readable/writable registers to make settings for such functions as self-test mode. ADCER0 and ADCER1 are initialized to H'0000 by a power-on reset or in hardware standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADR FMT	DIAG SEL	DIAG VALE	DIAG VAL2	DIAGM	DIAGLD	DIAGVAL[1:0]		CKS	-	-	-	-	-	-	ITT RGS*
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R/W

Note: * This bit is provided only in ADCER0 (ADC_A).
Bit 0 in ADCER1 (ADC_B) is reserved.

Bit	Bit Name	Initial Value	R/W	Description
15	ADRFMT	0	R/W	A/D Data Register Format Select The format of the A/D data register associated with a channel on which A/D-converted value addition mode is selected is fixed to left-alignment, irrespective of the ADRFMT bit value. For details on the format of the A/D data registers, see section 20.4.1, A/D Data Registers 0 to 27, 40 to 48, Diag0, and Diag1 (ADR0 to ADR27, ADR40 to ADR48, ADRD0, and ADRD1). 0: Selects left-alignment. 1: Selects right-alignment.
14	DIAGSEL	0	R/W	Self-Test Voltage Point Number Select Selects whether three voltages (AVref × 0, AVref × 1/2 and Avref × 1) or five voltages (AVref × 0, AVref × 1/4, AVref × 1/2, AVref × 3/4, and Avref × 1) are effective for conversion in the self-test process. 0: Three voltage values (AVref × 0, AVref × 1/2, and Avref × 1) are effective as voltages for conversion in the self-test process. 1: Five voltage values (AVref × 0, AVref × 1/4, AVref × 1/2, AVref × 3/4, and Avref × 1) are effective as voltages for conversion in the self-test process.

Bit	Bit Name	Initial Value	R/W	Description
13	DIAGVALE	0	R/W	DIAGVAL2 Valid Settings For the settings of self-test voltage, select whether the settings of DIAGVAL1 and 0 or of DIAGVAL2 are effective. 0: The settings of DIAGVAL1 and 0 are effective (Invalid the settings of DIAGVAL2). 1: The setting of DIAGVAL2 is effective (Invalid the settings of DIAGVAL1 and 0).
12	DIAGVAL2	0	R/W	Self-Test Voltage Select 2 When DIAGSEL bit is 1, DIAGM bit is 1, DIAGLD bit is 1, and DIAGVALE bit is 1. Select the voltage value whether AVref × 1/4 or AVref × 3/4 to convert in the self-test process. 0: Perform a self-test for AVref × 1/4 voltage value. 1: Perform a self-test for AVref × 3/4 voltage value.

Bit	Bit Name	Initial Value	R/W	Description
11	DIAGM	0	R/W	Self-Test Enable Self-test function is used to detect errors in the A/D converter (ADC_A and ADC_B). When DIAGSEL bit is 0 The ADC_A and ADC_B convert the internally generated voltage values AVref (AVrefh_A and AVrefh_B) × 0, AVref × 1/2 and AVref × 1. Upon completion of the conversion, A/D converted values and information on converted voltages are stored in A/D data registers Diag0 and Diag1 (ADRD0 and ADRD1). After that, ADRD0 and ADRD1 are read by software to determine whether the A/D converted values are in the normal range or not. A self-test is performed before the lowest-numbered channel is converted in the scan conversion process. In each execution of a self-test, one of the three voltage values is converted, and the three voltage values are automatically rotated each time a self-test is executed. The execution time for self-test is equal to the A/D conversion time for a channel. To prevent incorrect operation, the ADSCACT bit in ADREF must be cleared to 0 while the DIAGM value is changed. 0: Does not perform a self-test for the A/D converter. 1: One of three voltages is converted in self-testing by the A/D converter.

Bit	Bit Name	Initial Value	R/W	Description
11	DIAGM	0	R/W	When DIAGSEL bit is 1 The ADC_A and ADC_B convert the internally generated voltage values AVref (AVrefh_A and AVrefh_B) × 0, AVref × 1/4, AVref × 1/2,AVref × 3/4 and AVref × 1. Upon completion of the conversion, A/D converted values and information on converted voltages are stored in A/D data registers Diag0 and Diag1 (ADRD0 and ADRD1). After that, ADRD0 and ADRD1 are read by software to determine whether the A/D converted values are in the normal range or not. A self-test is performed before the lowest-numbered channel is converted in the scan conversion process. In each execution of a self-test, one of the five voltage values is converted, and the three voltage values are automatically rotated each time a self-test is executed. The execution time for self-test is equal to the A/D conversion time for a channel. To prevent incorrect operation, the ADSCACT bit in ADREF must be cleared to 0 while the DIAGM value is changed. 0: Does not perform a self-test for the A/D converter. 1: One of three voltages is converted in self-testing by the A/D converter.

Bit	Bit Name	Initial Value	R/W	Description
10	DIAGLD	0	R/W	Self-test Mode Select When DIAGSEL bit is 0 and DIAGM bit is 1 Selects whether the three voltage values that are converted in the self-test process are to be rotated or to be fixed. Writing 0 to the DIAGLD bit causes a conversion to be performed with rotating voltage values in the following sequence: Avref × 0 → Avref × 1/2 → Avref × 1. If a self-test is performed beginning with AVref × 0 upon power-on reset, the voltage does not return to AVref × 0 even after the completion of the scan conversion. If a scan conversion is executed again, rotation is resumed from where it ended in the previous conversion operation. Writing 1 to the DIAGLD bit causes a conversion at a fixed voltage value that is selected by the DIAGVAL1 and DIAGVAL0 bits in ADCER. (Automatic rotation is not performed.) Writing 0 again to the DIAGLD bit causes the start of rotation from a fixed voltage value (in a loading function). 0: Self-test is performed with automatic rotation. 1: Self-test is performed with the fixed voltage value selected by DIAGVALE, DIAGVAL2, DIAGVAL1 and DIAGVAL0.

Bit	Bit Name	Initial Value	R/W	Description
10	DIAGLD	0	R/W	When DIAGSEL bit is 1 and DIAGM bit is 1 Selects whether the five voltage values that are converted in the self-test process are to be rotated or to be fixed. Writing 0 to the DIAGLD bit causes a conversion to be performed with rotating voltage values in the following sequence: Avref × 0 → Avref × 1/4 → Avref × 1/2 → → Avref × 3/4 → Avref × 1. If a self-test is performed beginning with AVref × 0 upon power-on reset, the voltage does not return to AVref × 0 even after the completion of the scan conversion. If a scan conversion is executed again, rotation is resumed from where it ended in the previous conversion operation. Writing 1 to the DIAGLD bit causes a conversion at a fixed voltage value that is selected by the DIAGVALE, DIAGVAL2, DIAGVAL1 and DIAGVAL0 bits in ADCER. (Automatic rotation is not performed.) Writing 0 again to the DIAGLD bit causes the start of rotation from a fixed voltage value (in a loading function). 0: Self-test is performed with automatic rotation. 1: Self-test is performed with the fixed voltage value selected by DIAGVALE, DIAGVAL2, DIAGVAL1 and DIAGVAL0.
9, 8	DIAGVAL [1:0]	00	R/W	Self-Test Voltage Select For details, see the description of the DIAGLD bit. When the DIAGVALE bit is 1, settings of these bits are invalid. If the DIAGVALE bit is 0 and the value of these bits is B'00, execution of self-testing by writing 1 to the DIAGLD bit is disabled. 00: Reserved 01: Perform a self-test at a voltage AVref × 0 10: Perform a self-test at a voltage AVref × 1/2 11: Perform a self-test at a voltage AVref × 1

Bit	Bit Name	Initial Value	R/W	Description
7	CKS	0	R/W	Clock Select Selects A/D conversion time. To prevent incorrect operation, both the ADSCACT and ADITACT bits in ADREF must be 0 while changing the value of the CKS bit. 0: A/D conversion time = 50 states (based on Pϕ) 1: A/D conversion time = 25 states (based on Pϕ) (Setting prohibited when two-times multiplication has been set for the peripheral clock)
6 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	ITTRGS*	0	R/W	Expanded Interrupt Conversion Trigger Source Selection Selects the AN0 interrupt conversion trigger source between the timer D00A and timer G1 in the ATU-III. The ITTRGS bit becomes valid only when the ADTRGE0 bit in the ADTRE0 register is set to 1 and the ADTRS0 bit in the ADTRS0 register is cleared to 0. 0: AN0 interrupt conversion is triggered by the timer D00A in the ATU-III. 1: AN0 interrupt conversion is triggered by the timer G1 in the ATU-III.

Note: * The ITTRGS bit (bit 0) is provided only in ADCER0 (ADC_A). Bit 0 in ADCER1 (ADC_B) is reserved.

20.4.4 A/D Channel Select Registers 0, 1, and 3 (ADANS0, ADANS1, and ADANS3)

ADANS0, ADANS1, and ADANS3 are used to select channels that are subject to scan conversion. ADANS0, ADANS1, and ADANS3 are initialized to H'0000 by a power-on reset or in hardware standby mode. To prevent incorrect operation, the ADSCACT bit in ADREF must be cleared to 0 while the ADANS register values are changed.

Note: ADANS0, ADANS1, and ADANS3 are registers selects scan conversion channels; they are not used to select an interrupt conversion channel. An interrupt conversion channel is selected by the A/D interrupt trigger enable register (ADTRE).

If a channel is selected by both the ADANS and ADTRE registers, it is subject to conversion in both scan conversion and interrupt conversion. A channel that is selected only by the ADTRE register is excluded from the list of channels eligible for scan conversion, and only receives an interrupt conversion.

- ADANS0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ANS15	ANS14	ANS13	ANS12	ANS11	ANS10	ANS9	ANS8	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	ANS15 to ANS0	All 0	R/W	Setting the ANSn bit to 1 selects ANn. The correspondence between ANn and the ANSn bit is shown in table 20.2. 0: ANn is not selected. 1: ANn is selected.

Note: n = 0 to 15

- ADANS1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	ANS27	ANS26	ANS25	ANS24	ANS23	ANS22	ANS21	ANS20	ANS19	ANS18	ANS17	ANS16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 0	ANS27 to ANS16	All 0	R/W	Setting the ANSn bit to 1 selects ANn. The correspondence between ANn and the ANSn bit is shown in table 20.2. 0: ANn is not selected. 1: ANn is selected.

Note: n = 16 to 27

- ADANS3

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	ANS48	ANS47	ANS46	ANS45	ANS44	ANS43	ANS42	ANS41	ANS40
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8 to 0	ANS48 to ANS40	All 0	R/W	Setting the ANSn bit to 1 selects ANn. The correspondence between ANn and the ANSn bit is shown in table 20.2. 0: ANn is not selected. 1: ANn is selected.

Note: n = 40 to 48

20.4.5 A/D Conversion Status Registers 0 and 1 (ADREF0 and ADREF1)

ADREF0 and ADREF1 indicate the status of the A/D converter. ADREF0 and ADREF1 are initialized to H'00 by a power-on reset or in hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	ADS CACT	ADI TACT	-	-	-	-	-	ADF
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/(W)*

Note: * Only 0 can be written to clear the flag after reading the flag as 1.

Bit	Bit Name	Initial Value	R/W	Description
7	ADSCACT	0	R	Scan Conversion Status Indicates whether the scan conversion process is in the idle state or it is being executed. This is a read-only bit and cannot be written. If an interrupt conversion is started during a scan conversion, the A/D converter stops the scan conversion process and preferentially executes the interrupt conversion. However, until such time that all scan conversion is completed, the ADSCACT bit maintains to be set to 1 and is not cleared to 0. 0: Indicates that the Scan conversion process is in idle state. 1: Indicates that the Scan conversion process is being executed.
6	ADITACT	0	R	Interrupt Conversion Status Indicates whether the interrupt conversion process is in the idle state or it is being executed. This is a read-only bit and cannot be written. The ADSCACT and ADITACT bits can indicate the status of the ADC_A and ADC_B. For details, see table 20.4. 0: Indicates that the interrupt conversion process is in idle state. 1: Indicates that the Interrupt conversion process is being executed.

Bit	Bit Name	Initial Value	R/W	Description
5 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	ADF	0	R/(W)*	Single Scan End Flag This bit is set to 1 each time scanning ends in the scan conversion process (when all selected channels are converted). 1 cannot be written to this bit. When the ADF bit is set to 1, either a scan conversion end interrupt or a DMA transfer request to the DMAC can be generated. In this manner, processing such as storing the contents of the A/D data register to the RAM can be implemented by means of either software or the DMAC. 0: Indicates that the scan conversion process is in idle state. 1: Indicates that a single scan has been completed and the A/D-converted values on all selected ANn channels have been transferred to ADRn. [Clearing conditions] • 0 is written to this bit after reading 1. • The DMAC is started-up by the ADI. [Setting condition] • All analog conversion has been completed during each scanning in scan conversion process.

Note: * Only 0 can be written to clear the flag after reading the flag as 1.

Table 20.4 Relationship between ADC_A and ADC_B Status and ADSCACT and ADITACT

ADSCACT	ADITACT	ADC_A and ADC_B Status	Source of Scan Conversion	Source of Interrupt Conversion
0	0	Idle state	No	No
	1	Interrupt conversion	No	Yes
1	0	Scan conversion	Yes	No
	1	Interrupt conversion	Yes	Yes

20.4.6 A/D-Converted Value Addition Mode Select Registers 0 to 5 (ADADS0 to ADADS5)

ADADS0 to ADADS5 select ANn (n = 0 to 27, 40 to 48) on which A/D conversion is performed successively 2 to 4 times and then converted values are added (integrated). ADADS0 to ADADS5 are initialized to H'00 by a power-on reset or in hardware standby mode.

- ADADS0

Bit:	7	6	5	4	3	2	1	0
	ADS7	ADS6	ADS5	ADS4	ADS3	ADS2	ADS1	ADS0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	ADS7 to ADS0	All 0	R/W	A/D-Converted Value Addition Channel Select When the ADSn bit is set to 1, the A/D converter performs conversion on ANn successively 2 to 4 times and returns the added (integrated) conversion results to the A/D data register. If the ADSn bit is cleared to 0, the A/D converter performs a normal 1-time conversion of ANn and returns the conversion result to the A/D data register. Irrespective of whether a scan conversion or an interrupt conversion is to be performed, the A/D converter determines whether or not to perform an addition according to the ADSn value. To prevent incorrect operation, both the ADSCACT and ADITACT bits in ADREF must be cleared to 0 while the ADSn bit value is changed. The correspondence between ANn and the ANSn bit is shown in table 20.2. How to select the addition count is described in section 20.4.7, A/D-Converted Value Addition Count Select Registers 0 and 1 (ADADC0 and ADADC1). 0: A/D-converted value addition mode is not selected. 1: A/D-converted value addition mode (successively 2 to 4 times of addition) is selected.

Note: n = 0 to 7

- ADADS1

Bit:	7	6	5	4	3	2	1	0
	ADS47	ADS46	ADS45	ADS44	ADS43	ADS42	ADS41	ADS40
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	ADS47 to ADS40	All 0	R/W	A/D-Converted Value Addition Channel Select When the ADSn bit is set to 1, the A/D converter performs conversion on ANn successively 2 to 4 times and returns the added (integrated) conversion results to the A/D data register. If the ADSn bit is cleared to 0, the A/D converter performs a normal 1-time conversion of ANn and returns the conversion result to the A/D data register. Irrespective of whether a scan conversion or an interrupt conversion is to be performed, the A/D converter determines whether or not to perform an addition according to the ADSn value. To prevent incorrect operation, both the ADSCACT and ADITACT bits in ADREF must be cleared to 0 while the ADSn bit value is changed. The correspondence between ANn and the ANSn bit is shown in table 20.2. How to select the addition count is described in section 20.4.7, A/D-Converted Value Addition Count Select Registers 0 and 1 (ADADC0 and ADADC1). 0: A/D-converted value addition mode is not selected. 1: A/D-converted value addition mode (successively 2 to 4 times of addition) is selected.

Note: n = 40 to 47

- ADADS2

Bit:	7	6	5	4	3	2	1	0
	ADS15	ADS14	ADS13	ADS12	ADS11	ADS10	ADS9	ADS8
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	ADS15 to ADS8	All 0	R/W	A/D-Converted Value Addition Channel Select When the ADSn bit is set to 1, the A/D converter performs conversion on ANn successively 2 to 4 times and returns the added (integrated) conversion results to the A/D data register. If the ADSn bit is cleared to 0, the A/D converter performs a normal 1-time conversion of ANn and returns the conversion result to the A/D data register. Irrespective of whether a scan conversion or an interrupt conversion is to be performed, the A/D converter determines whether or not to perform an addition according to the ADSn value. To prevent incorrect operation, both the ADSCACT and ADITACT bits in ADREF must be cleared to 0 while the ADSn bit value is changed. The correspondence between ANn and the ANSn bit is shown in table 20.2. How to select the addition count is described in section 20.4.7, A/D-Converted Value Addition Count Select Registers 0 and 1 (ADADC0 and ADADC1). 0: A/D-converted value addition mode is not selected. 1: A/D-converted value addition mode (successively 2 to 4 times of addition) is selected.

Note: n = 8 to 15

- ADADS3

Bit:	7	6	5	4	3	2	1	0
	ADS23	ADS22	ADS21	ADS20	ADS19	ADS18	ADS17	ADS16
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	ADS23 to ADS16	All 0	R/W	A/D-Converted Value Addition Channel Select When the ADSn bit is set to 1, the A/D converter performs conversion on ANn successively 2 to 4 times and returns the added (integrated) conversion results to the A/D data register. If the ADSn bit is cleared to 0, the A/D converter performs a normal 1-time conversion of ANn and returns the conversion result to the A/D data register. Irrespective of whether a scan conversion or an interrupt conversion is to be performed, the A/D converter determines whether or not to perform an addition according to the ADSn value. To prevent incorrect operation, both the ADSCACT and ADITACT bits in ADREF must be cleared to 0 while the ADSn bit value is changed. The correspondence between ANn and the ANSn bit is shown in table 20.2. How to select the addition count is described in section 20.4.7, A/D-Converted Value Addition Count Select Registers 0 and 1 (ADADC0 and ADADC1). 0: A/D-converted value addition mode is not selected. 1: A/D-converted value addition mode (successively 2 to 4 times of addition) is selected.

Note: n = 16 to 23

- ADADS4

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	ADS27	ADS26	ADS25	ADS24
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
3 to 0	ADS27 to ADS24	All 0	R/W	A/D-Converted Value Addition Channel Select When the ADSn bit is set to 1, the A/D converter performs conversion on ANn successively 2 to 4 times and returns the added (integrated) conversion results to the A/D data register. If the ADSn bit is cleared to 0, the A/D converter performs a normal 1-time conversion of ANn and returns the conversion result to the A/D data register. Irrespective of whether a scan conversion or an interrupt conversion is to be performed, the A/D converter determines whether or not to perform an addition according to the ADSn value. To prevent incorrect operation, both the ADSCACT and ADITACT bits in ADREF must be cleared to 0 while the ADSn bit value is changed. The correspondence between ANn and the ANSn bit is shown in table 20.2. How to select the addition count is described in section 20.4.7, A/D-Converted Value Addition Count Select Registers 0 and 1 (ADADC0 and ADADC1). 0: A/D-converted value addition mode is not selected. 1: A/D-converted value addition mode (successively 2 to 4 times of addition) is selected.

Note: n = 24 to 27

- ADADS5

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	ADS48
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
0	ADS48	0	R/W	A/D-Converted Value Addition Channel Select When the ADSn bit is set to 1, the A/D converter performs conversion on ANn successively 2 to 4 times and returns the added (integrated) conversion results to the A/D data register. If the ADSn bit is cleared to 0, the A/D converter performs a normal 1-time conversion of ANn and returns the conversion result to the A/D data register. Irrespective of whether a scan conversion or an interrupt conversion is to be performed, the A/D converter determines whether or not to perform an addition according to the ADSn value. To prevent incorrect operation, both the ADSCACT and ADITACT bits in ADREF must be cleared to 0 while the ADSn bit value is changed. The correspondence between ANn and the ANSn bit is shown in table 20.2. How to select the addition count is described in section 20.4.7, A/D-Converted Value Addition Count Select Registers 0 and 1 (ADADC0 and ADADC1). 0: A/D-converted value addition mode is not selected. 1: A/D-converted value addition mode (successively 2 to 4 times of addition) is selected.

Note: n = 48

Figure 20.2 shows a scan conversion sequence in which both the ADS42 and ADS46 bits are set to 1, based on the assumption that the addition count is set to 4, and that channels AN40 to AN47 are selected. The conversion process begins with AN40. The AN42 conversion is performed successively 4 times, and the addition (integration) value is returned to the data register, after which the AN43 conversion process is started. If an interrupt conversion is requested in the midst of a scan conversion, the scan conversion process is stopped and an A/D conversion is preferentially executed on the channel in which an interrupt conversion was requested. Upon completion of the interrupt conversion, the scan conversion process is resumed from the A/D conversion on the interrupted channel. However, if the ADSn bit of the interrupted channel (ANn) is set to 1, even if addition has been performed at least once (two to four times), the conversion is restarted from the 1st conversion.

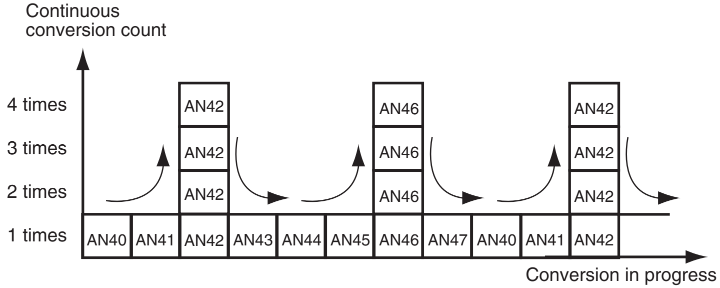


Figure 20.2 Scan Conversion Sequence with ADS42 and ADS46 Bits = 1

20.4.7 A/D-Converted Value Addition Count Select Registers 0 and 1 (ADADC0 and ADADC1)

ADADC0 and ADADC1 set the addition count for channels for which A/D-converted value addition mode is selected. ADADC0 and ADADC1 are initialized to H'00 by a power-on reset or in hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	ADC[1:0]	
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	ADC[1:0]	00	R/W	Addition Count Select These bits select the number of additions to be performed in A/D-converted value addition mode. These bits have no effect on the A/D conversion of channels for which A/D-converted value addition mode is not selected. To prevent incorrect operation, both the ADSCACT and ADITACT bits in ADREF must be cleared to 0 while the ADC1 and ADC0 bit values are changed. 00: 1-time conversion (normal conversion) 01: 2-time conversion 10: 3-time conversion 11: 4-time conversion

20.4.8 A/D Interrupt Trigger Enable Registers 0 and 1 (ADTRE0 and ADTRE1)

ADTRE0 and ADTRE1 enable or disable an interrupt conversion request for AN0 to AN15 and AN40 to AN47. Channels for which interrupt conversion is enabled are subjected to an interrupt conversion when a corresponding interrupt conversion request is input. ADTRE0 and ADTRE1 are initialized to H'0000 and H'00, respectively, by a power-on reset or in hardware standby mode.

- ADTRE0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADTR GE15	ADTR GE14	ADTR GE13	ADTR GE12	ADTR GE11	ADTR GE10	ADTR GE9	ADTR GE8	ADTR GE7	ADTR GE6	ADTR GE5	ADTR GE4	ADTR GE3	ADTR GE2	ADTR GE1	ADTR GE0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	ADTRGE15 to ADTRGE0	All 0	R/W	Interrupt Conversion Request Enable Setting the ADTRGEn bit to 1 enables the interrupt conversion request to the corresponding ANn channel. The correspondence among ANn, the ADTRGEn bit and the interrupt request trigger source is shown in table 20.2. 0: Disables an interrupt conversion request to ANn by ATU-III timer or software trigger (ADSTRGn). 1: Enables an interrupt conversion request to ANn by ATU-III timer or software trigger (ADSTRGn).

Note: n = 0 to 15

- ADTRE1

Bit:	7	6	5	4	3	2	1	0
	ADTRGE47	ADTRGE46	ADTRGE45	ADTRGE44	ADTRGE43	ADTRGE42	ADTRGE41	ADTRGE40
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	ADTRGE47 to ADTRGE40	All 0	R/W	Interrupt Conversion Request Enable Setting the ADTRGEn bit to 1 enables the interrupt conversion request to the corresponding ANn channel. The correspondence among ANn, the ADTRGEn bit and the interrupt request trigger source is shown in table 20.2. If the timer G0 signal is input when both the ADTRGE40 and ADTRGE44 bits in ADTRE1 are set to 1 and the timer G0 in the ATU-III is selected as an interrupt request source, the A/D converter converts AN44 after converting AN40. Similar operations can also be performed by the interrupt conversion of AN41 and AN45 on timer G1, of AN42 and AN46 on timer G2, and of AN43 and AN47 on timer G3. 0: Disables an interrupt conversion request to ANn by ATU-III timer or software trigger (ADSTRGn). 1: Enables an interrupt conversion request to ANn by ATU-III timer or software trigger (ADSTRGn).

Note: n = 40 to 47

20.4.9 A/D Interrupt Trigger Source Select Registers 0 and 1 (ADTRS0 and ADTRS1)

ADTRS0 and ADTRS1 select the trigger source for interrupt conversion. Either an ATU-III timer trigger or a software trigger caused by writing to the ADSTRG0 and ADSTRG1 registers can be selected. ADTRS0 and ADTRS1 are initialized to H'0000 and H'00 respectively by a power-on reset or in hardware standby mode.

- ADTRS0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADT RS15	ADT RS14	ADT RS13	ADT RS12	ADT RS11	ADT RS10	ADT RS9	ADT RS8	ADT RS7	ADT RS6	ADT RS5	ADT RS4	ADT RS3	ADT RS2	ADT RS1	ADT RS0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 1	ADTRS15 to ADTRS1	All 0	R/W	Interrupt Conversion Trigger Source Select If the ADTRSn bit is cleared to 0 and the ADTREn bit in the A/D interrupt trigger enable register is set to 1, the A/D converter detects an edge and begins the interrupt conversion on ANn when an ATU-III timer trigger is input. If the ADTRSn bit is set to 1, the A/D converter detects an edge and begins the interrupt conversion on ANn when 1 is written to the ADSTRGn bit in the A/D interrupt software trigger register. The correspondence among the ADTRSn bit, ANn, and the interrupt request trigger source is shown in table 20.2. 0: Uses the timer D in the ATU-III as an ANn interrupt conversion request source. 1: Uses a software trigger (ADSTRGn) as an ANn interrupt conversion request source.

Bit	Bit Name	Initial Value	R/W	Description
0	ADTRS0	0	R/W	Interrupt Conversion Trigger Source Select For AN0, either the timer D00A or timer G1 can be used as an ATU-III timer trigger. If the ADTRS0 bit is cleared to 0 and the ITTRGS bit in ADCER0 is cleared to 0, the timer D00A is selected as an interrupt conversion request trigger source. If the ADTRS0 bit is cleared to 0 and the ITTRGS bit is set to 1, the timer G1 is selected as an interrupt conversion request trigger source. 0: Uses the timer D00A or timer G1 in the ATU-III as an AN0 interrupt conversion request source. 1: Uses the software trigger (ADSTRGn) as an AN0 interrupt conversion request source.

Note: n = 1 to 15

- ADTRS1

Bit:	7	6	5	4	3	2	1	0
	ADT RS47	ADT RS46	ADT RS45	ADT RS44	ADT RS43	ADT RS42	ADT RS41	ADT RS40
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	ADTRS47 to ADTRS40	All 0	R/W	Interrupt Conversion Trigger Source Select If the ADTRSn bit is cleared to 0 and the ADTREn bit in the A/D interrupt trigger enable register is set to 1, the A/D converter detects an edge and begins the interrupt conversion on ANn when an ATU-III timer trigger is input. If the ADTRSn bit is set to 1, the A/D converter detects an edge and begins the interrupt conversion on ANn when 1 is written to the ADSTRGn bit in the A/D interrupt software trigger register. The correspondence among the ADTRSn bit, ANn, and the interrupt request trigger source is shown in table 20.2. 0: Uses the timer G in ATU-III as an ANn interrupt conversion request source. 1: Uses the software trigger (ADSTRGn) as an ANn interrupt conversion request source.

Note: n = 40 to 47

20.4.10 A/D Interrupt Software Trigger Registers 0 and 1 (ADSTRG0 and ADSTRG1)

ADSTRG0 and ADSTRG1 start an interrupt conversion by software. ADSTRG0 and ADSTRG1 are write-only registers and they are always read as 0s.

- ADSTRG0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADST RG15	ADST RG14	ADST RG13	ADST RG12	ADST RG11	ADST RG10	ADST RG9	ADST RG8	ADST RG7	ADST RG6	ADST RG5	ADST RG4	ADST RG3	ADST RG2	ADST RG1	ADST RG0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	ADSTRG15 to ADSTRG0	All 0	W	Interrupt Conversion Software Trigger If 1 is written to the ADSTRGn bit when the ADTRSn bit in the A/D interrupt trigger source select register associated with the ANn is set to 1 and the ADTREn bit in the A/D interrupt trigger enable register is set to 1, the A/D converter detects an edge and starts the interrupt conversion of ANn. If an ANn is not subject to an interrupt conversion request, 0 should be written to it. ANn channels for which 0s are written are not affected by any of these operations. If an interrupt conversion is requested, the interrupt source is stored in the internal circuit in units of ANn. If an interrupt conversion on ANn with a source is performed and completed, the source associated with the ANn is cleared. Consequently, if 1 is written to the ADSTRGn and subsequently 0 is written to it, the source associated with the ANn is not cleared, and, therefore, the interrupt conversion is executed. It should be noted, however, that if a source is pending on ANn and then 1 is written to the ADSTRGn, it does not follow that an interrupt conversion is performed on the ANn twice. There is one source per channel. The same rule also applies to the execution of an interrupt conversion in response to a request from an ATU-III timer trigger. See table 20.2 for correspondence between ADSTRGn and ANn. 0: No interrupt conversion is request (software trigger) on the ANn. 1: Interrupt conversion is requested (software trigger) on the ANn.

Note: n = 0 to 15

- ADSTRG1

Bit:	7	6	5	4	3	2	1	0
	ADST RG47	ADST RG46	ADST RG45	ADST RG44	ADST RG43	ADST RG42	ADST RG41	ADST RG40
Initial value:	0	0	0	0	0	0	0	0
R/W:	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	ADSTRG47 to ADSTRG40	All 0	W	Interrupt Conversion Software Trigger If 1 is written to the ADSTRGn bit when the ADTRSn bit in the A/D interrupt trigger source select register associated with the ANn is set to 1 and the ADTREn bit in the A/D interrupt trigger enable register is set to 1, the A/D converter detects an edge and starts the interrupt conversion of ANn. If an ANn is not subject to an interrupt conversion request, 0 should be written to it. ANn channels for which 0s are written are not affected by any of these operations. If an interrupt conversion is requested, the interrupt source is stored in the internal circuit in units of ANn. If an interrupt conversion on ANn with a source is performed and completed, the source associated with the ANn is cleared. Consequently, if 1 is written to the ADSTRGn and subsequently 0 is written to it, the source associated with the ANn is not cleared, and, therefore, the interrupt conversion is executed. It should be noted, however, that if a source is pending on ANn and then 1 is written to the ADSTRGn, it does not follow that an interrupt conversion is performed on the ANn twice. There is one source per channel. The same rule also applies to the execution of an interrupt conversion in response to a request from an ATU-III timer trigger. See table 20.2 for correspondence between ADSTRGn and ANn. 0: No interrupt conversion is request ed(software trigger) on the ANn. 1: Interrupt conversion is requested (software trigger) on the ANn.

Note: n = 40 to 47

20.4.11 A/D Interrupt Trigger Conversion End Flag Registers 0 and 1 (ADTRF0 and ADTRF1)

ADTRF0 and ADTRF1 indicate that an interrupt conversion has been completed. When an interrupt conversion has been completed, the ADTFn bit corresponding to the channel (ANn) is set to 1. ADTRF0 and ADTRF1 are initialized to H'0000 and H'00 respectively by a power-on reset or in hardware standby mode.

- ADTRF0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADT F15	ADT F14	ADT F13	ADT F12	ADT F11	ADT F10	ADT F9	ADT F8	ADT F7	ADT F6	ADT F5	ADT F4	ADT F3	ADT F2	ADT F1	ADT F0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only 0 can be written to clear the flag after reading the flag as 1.

Bit	Bit Name	Initial Value	R/W	Description
15 to 1	ADTF15 to ADTF1	All 0	R/(W)*	Interrupt Conversion End Flag ADTFn is a status flag bit which indicates that an interrupt conversion has been completed. 1 must not be written to ADTFn. When ADTFn is set to 1, an ANn interrupt conversion end interrupt (ADIDn) can be generated. See table 20.2 for correspondence between ADTFn and ANn. 0: Indicates that an interrupt conversion process on ANn is in idle state. 1: Indicates that an interrupt conversion process on ANn has been completed and the conversion result has been transferred to ADRn. [Clearing condition] • 0 is written to this bit after reading 1. [Setting condition] • An interrupt conversion process on ANn has been completed.

Bit	Bit Name	Initial Value	R/W	Description
0	ADTF0	0	R/(W)*	Interrupt Conversion End Flag ADTF0 is a status flag bit which indicates that an interrupt conversion has been completed. 0 must not be written to ADTF0. When ADTF0 is set to 1, AN0 interrupt conversion end interrupt (ADID0) can be generated. See table 20.2 for correspondence between ADTF0 and AN0. 0: Indicates that an interrupt conversion process on AN0 is in idle state. 1: Indicates that an interrupt conversion process on AN0 has been completed and the conversion result has been transferred to ADR0. [Clearing conditions] • 0 is written to this bit after reading 1. • A-DMAC is started-up by ADID0. [Setting condition] • An interrupt conversion process on AN0 has been completed.

Notes: * Only 0 can be written to clear the flag after reading the flag as 1.

1. Even when the ADTFn is not cleared to 0, an interrupt conversion request on ANn can be accepted. Storing timing on an A/D data register n should be provided with care.
2. n = 1 to 15

- ADTRF1

Bit:	7	6	5	4	3	2	1	0
	ADT F47	ADT F46	ADT F45	ADT F44	ADT F43	ADT F42	ADT F41	ADT F40
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only 0 can be written to clear the flag after reading the flag as 1.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	ADTF47 to ADTF40	All 0	R/(W)*	Interrupt Conversion End Flag ADTFn is a status flag bit which indicates that an interrupt conversion has been completed. 0 must not be written to ADTFn. When ADTFn is set to 1, an ANn interrupt conversion end interrupt (ADIDn) can be generated. ANn can generate a DMA transfer request to the DMAC. See table 20.2 for correspondence between ADTFn and ANn. 0: Indicates that an interrupt conversion process on ANn is in idle state. 1: Indicates that an interrupt conversion process on ANn has been completed and the conversion result has been transferred to ADRn. [Clearing conditions] • 0 is written to this bit after reading 1. • A-DMAC is started-up by ADIDn. [Setting condition] • An interrupt conversion process on ANn has been completed.

Notes: * Only 0 can be written to clear the flag after reading the flag as 1.

1. Even when the ADTFn is not cleared to 0, an interrupt conversion request on ANn can be accepted. Storing timing on an A/D data register n should be provided with care.
2. n = 40 to 47

20.4.12 A/D Interrupt Trigger Conversion Interrupt Enable Registers 0 and 1 (ADTRD0 and ADTRD1)

ADTRD0 and ADIRD1 enable or disable an A/D interrupt conversion end interrupt generation when the ADTF bit in ADTRF is set to 1. ADTRD0 and ADTRD1 are initialized to H'0000 and H'00 respectively by a power-on reset or in hardware standby mode.

- ADTRD0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADI DE15	ADI DE14	ADI DE13	ADI DE12	ADI DE11	ADI DE10	ADI DE9	ADI DE8	ADI DE7	ADI DE6	ADI DE5	ADI DE4	ADI DE3	ADI DE2	ADI DE1	ADI DE0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 1	ADIDE15 to ADIDE1	All 0	R/W	Interrupt Conversion End Interrupt Enable The ADIDEn bit enables or disables an ANn interrupt conversion end interrupt (ADIDn) to be generated. To prevent incorrect operation, both the ADITACT bit in ADREF0 must be cleared to 0 while the ADIDEn bit value is changed. If the ADIDEn bit is 1 when the ADTFn bit of the interrupt conversion end flag register is set to 1 upon completion of ANn interrupt conversion, the ADIDn signal is generated. The ADIDn signal can be cleared by clearing ADTFn or ADIDEn to 0. The correspondence among the ADIDEn bit, ANn, and ADIDn is shown in table 20.2. 0: Disables an interrupt request upon completion of ANn interrupt conversion (ADIDn). 1: Enables an interrupt request upon completion of ANn interrupt conversion (ADIDn).

Bit	Bit Name	Initial Value	R/W	Description
0	ADIDE0	0	R/W	Interrupt Conversion End Interrupt Enable The ADIDE0 bit enables or disables an AN0 interrupt conversion end interrupt (ADID0) to be generated. To prevent incorrect operation, the ADITACT bit in ADREF0 must be 0 while the ADIDE0 bit is changed. If the ADIDE0 bit is 1 when the ADTF0 bit of the interrupt conversion end flag register is set to 1 upon completion of AN0 interrupt conversion, the ADID0 signal is generated. The ADID0 signal can be cleared by clearing ADTF0 or ADIDE0 to 0. Further, the AN0 can perform a DMA transfer by means of the A-DMAC based on ADID0. See table 20.2 for correspondence between ADIDE0, AN0, and ADID0. 0: Disables an interrupt request upon completion of AN0 interrupt conversion (ADID0) or a DMA transfer request. 1: Enables an interrupt request upon completion of AN0 interrupt conversion (ADID0) or a DMA transfer request.

Note: n = 1 to 15

- ADTRD1

Bit:	7	6	5	4	3	2	1	0
	ADI DE47	ADI DE46	ADI DE45	ADI DE44	ADI DE43	ADI DE42	ADI DE41	ADI DE40
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	ADIDE47 to ADIDE40	All 0	R/W	Interrupt Conversion End Interrupt Enable The ADIDEn bit enables or disables an ANn interrupt conversion end interrupt (ADIDn) to be generated. To prevent incorrect operation, the ADITACT bit in ADREF1 must be 0 while the ADIDEn bit is changed. If the ADIDEn bit is 1 when the ADTFn bit in the interrupt conversion end flag register is set to 1 upon completion of ANn interrupt conversion, the ADIDn signal is generated. The ADIDn signal can be cleared by clearing ADTFn or ADIDEn to 0. Further, the AN40 to AN47 can perform DMA transfers by means of the DMAC based on ADID40 to ADID47. See table 20.2 for correspondence between ADIDEn, ANn, and ADIDn. 0: Disables an interrupt request upon completion of ANn interrupt conversion (ADIDn) or an DMA transfer request. 1: Enables an interrupt request upon completion of ANn interrupt conversion (ADIDn) or an DMA transfer request.

Note: n = 40 to 47

20.4.13 Interface with CPU

The A/D data register is a 16-bit register. The peripheral bus connected to the CPU is 16 bits. The A/D data register must be read in units of words (16 bits). If the A/D data register is read in byte units by dividing a word into upper and lower bytes and performing read operations twice on it, the A/D converted value read in the first read operation and that read in the second read operation may change. To avoid this error, the A/D data register should not be read in byte units.

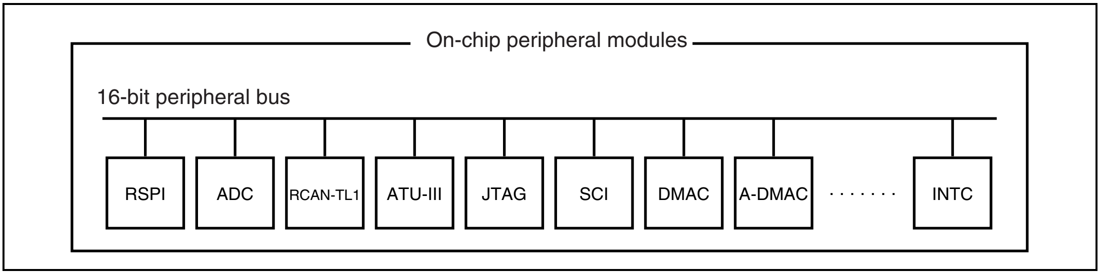


Figure 20.3 Interface between CPU and A/D Converter (ADC)

20.5 Operation

20.5.1 Scan Conversion

A scan conversion is performed in two operating modes: the single-cycle scan mode and continuous scan mode. In the single-cycle scan mode, one or more specified channels are scanned once. In the continuous scan mode, one or more specified channels are scanned until the ADST bit is cleared to 0 (changed from 1 to 0) by software.

The single-cycle scan mode is selected by clearing the ADCS bit in ADCSR to 0, while the continuous scan mode is selected by setting the ADCS bit to 1. A scan conversion is performed in ascending order of channel number (ADC_A: AN0 → AN27, ADC_B: AN40 → AN48).

In single-cycle scan mode, after scanning all selected channels once, the A/D converter sets the ADF bit in ADREF to 1 and then clears the ADSCACT bit in ADREF to 0 to completes the scan conversion. In continuous scan mode, after scanning all selected channels once, the A/D converter sets the ADF bit to 1 and then continues to scanning. The ADF bit is set to 1 each time scanning on specified channels is completed.

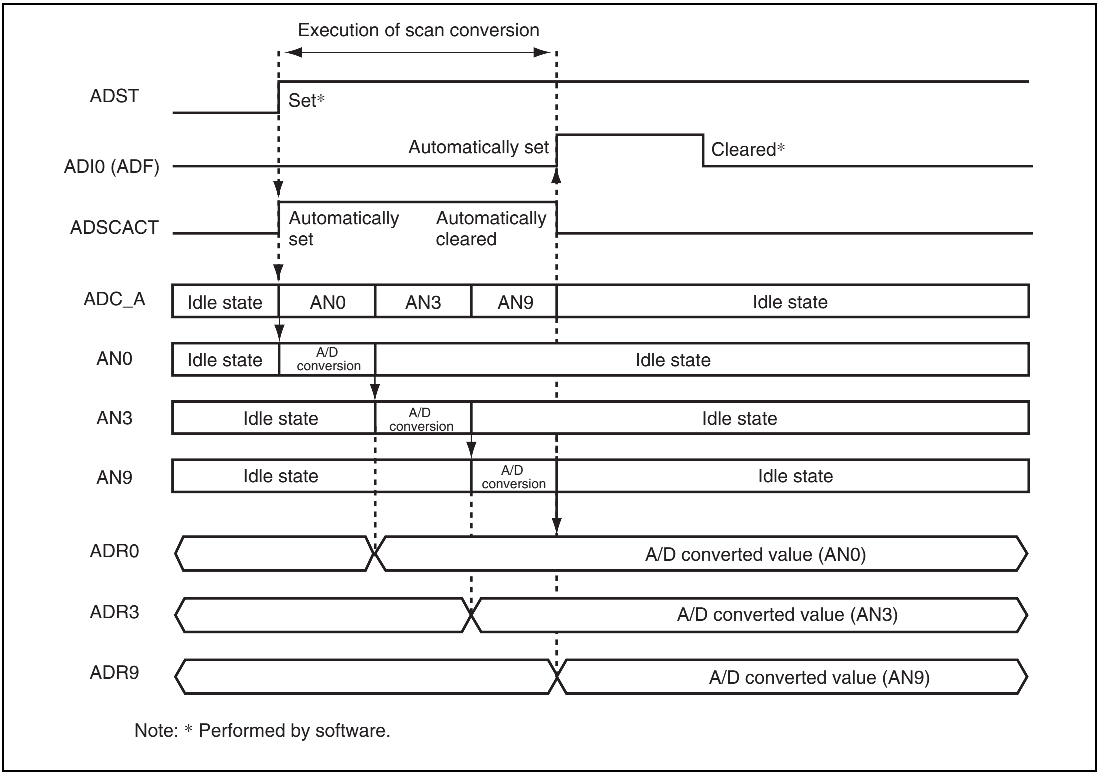
To stop the scanning, write 0 to the ADST bit when it is 1. Writing 0 to the ADST bit when it is 0 does not affect the A/D converter. Similarly, writing 1 to the ADST bit when it is 1 does not affect the A/D converter. Therefore, to stop a scan conversion started by a request other than the ADST bit, first write 1 to the ADST bit and then write 0 to it.

When the ADF bit is set to 1 while the ADIE bit in ADCSR is set to 1, an ADI interrupt request is generated. To clear the ADF bit to 0, write 0 to the ADF bit after reading it as 1. When the DMAC is started by an ADI interrupt, the ADF bit is automatically cleared to 0 and the ADI interrupt is also cleared.

20.5.2 Single-Cycle Scan Conversion Mode

The following is an example operation of single-scan conversion where three channels AN0, AN3, and AN9 are selected and an ADI0 interrupt is enabled. The same operations can also apply to ADC_B.

1. Clear the ADCS bit in A/D control register 0 (ADCSR0) to 0 and set the ADIE bit in ADCSR0 to 1.
2. Set bits ANS0, ANS3 and ANS9 in the A/D channel select register 0 (ADANS0).
3. Set the ADST bit in A/D control register 0 (ADSCR0) to 1 to start scan conversion. If the ADST bit is already set to 1, write 1 to it after clearing it to 0.
4. Starting the scan conversion sets the ADSCACT bit to 1. Then, the A/D conversion on channel AN0 is started. On completion of the A/D conversion, the A/D converted value is transferred to ADR0. After that, channels AN3 and AN9 are scanned in the order in the same way as in AN0.
5. When the A/D converted values of all the selected channels (AN0, AN3, and AN9) have been transferred to ADRn, the ADF bit is set to 1. At this time, an ADI0 interrupt is generated since the ADIE bit is set to 1. The ADSCACT bit is cleared to 0 and the scan conversion is completed.
6. Next, the ADI0 interrupt handler is started. In the interrupt handler, clear bit ADI0 by writing 0 to the ADF bit after reading it as 1. After that, read the contents of ADR0, ADR3, and ADR9.
7. Complete the ADI0 interrupt handler.



Note: * Performed by software.

Figure 20.4 Example Operation in Single-Cycle Scan Mode

20.5.3 Continuous Scan Conversion Mode

The following is an example operation of continuous scan conversion where three channels AN0, AN3, and AN9 are selected and an ADI0 interrupt is enabled. The same operations can also apply to ADC_B.

1 Set the ADCS bit and ADIE bit in A/D control register 0 (ADCSR0) to 1.

2. Set bits ANS0, ANS3 and ANS9 bits in A/D channel select register 0 (ADANS0).
3. Set the ADST bit in A/D control register 0 (ADSCR0) to 1 to start scan conversion. If the ADST bit is already 1, write 1 to it after clearing it to 0.
4. Starting the scan conversion sets the ADSCACT bit to 1. Then, the A/D conversion on channel AN0 is started. On completion of the A/D conversion, the A/D converted value is transferred to ADR0. After that, channels AN3 and AN9 are scanned in the order in the same way as in AN0.
5. When the A/D converted values of all the selected channels (AN0, AN3, and AN9) have been transferred to ADRn, the ADF bit is set to 1. At this time, an ADI0 interrupt is generated since the ADIE bit is set to 1. Also, the scan conversion returns to the start.
6. The ADI0 interrupt handler is started simultaneously. In the interrupt handler, clear bit ADI0 by writing 0 to the ADF bit after reading it as 1. After that, read the contents of ADR0, ADR3, and ADR9.
7. Complete the ADI0 interrupt handler.
8. Steps 4 to 7 are repeated as long as the ADST bit is 1. Clearing the ADST bit to 0 clears the ADSCACT bit to 0, and completes the scan conversion. Setting the ADST bit to 1 initiates scan conversion.

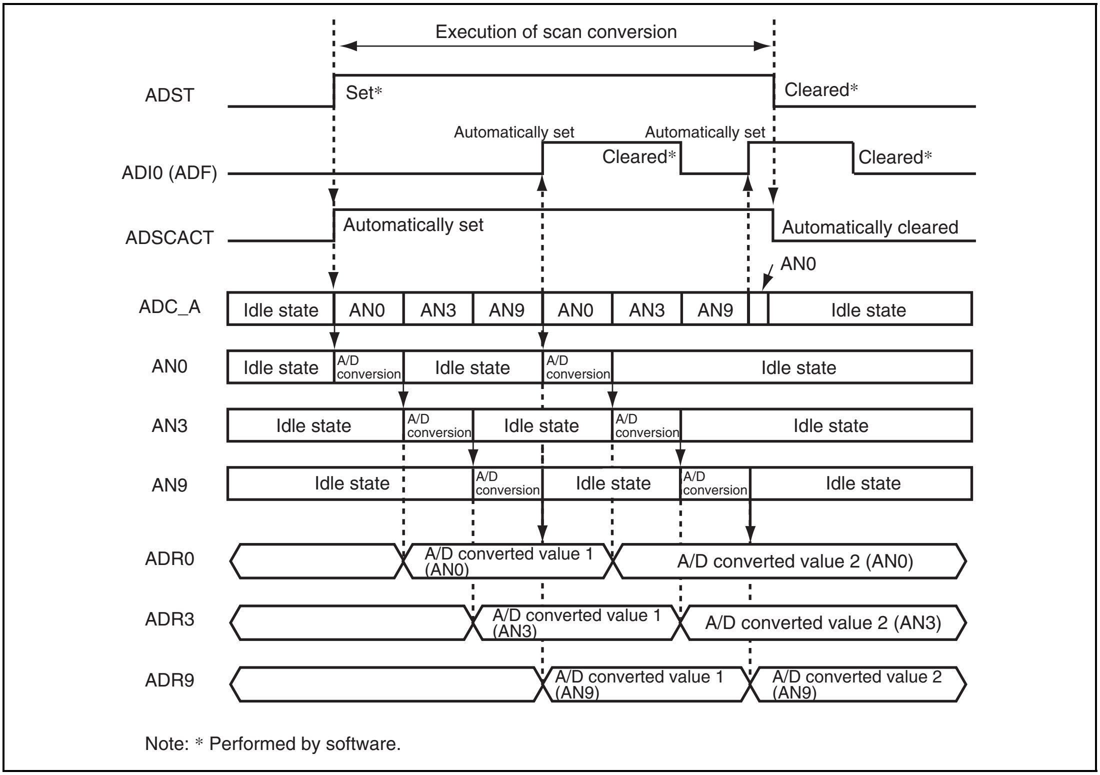


Figure 20.5 Example Operation in Continuous Scan Mode

20.5.4 Interrupt Conversion

When an ATU-III timer trigger or software trigger is requested on channels AN0 to AN15 and AN40 to AN47, A/D conversion is performed on the requested channels. Whereas the scan conversion converts all the selected channels for a request, the interrupt conversion converts all the channels selected by a request.

To perform interrupt conversion, set the ADTRGE bit in ADTRE to 1 and select the trigger source by the ADTRS bit in ADTRS. When interrupt conversion is requested by the selected trigger source, A/D conversion is performed on the corresponding AN channel. On completion of the interrupt conversion on the AN channel, the ADTF bit in ADTRF is set to 1. The ADTF bit is set to 1 each time an interrupt conversion is performed on an AN channel. Furthermore, if any interrupt conversion is performed, the ADITACT bit in ADREF is set to 1. When A/D conversion has been completed on all AN channels to which interrupt conversion is requested, the ADITACT bit is cleared to 0.

When interrupt conversion requests conflict, A/D conversion is performed according to the priority. ADC_A is prioritized as AN0 > AN1 > … AN14 > AN15, that is, the lower channel number corresponds to the higher priority. ADC_B is prioritized as AN40 > AN44 > AN41 > AN45 > AN42 > AN46 > AN43 > AN47, thus channel AN40 given the highest priority, and AN47 given the lowest priority. Note that this priority in the ADC_B channels is not in ascending order of channel numbers. When interrupt conversion is requested on other channels (ANj and ANk) during the interrupt conversion on channel ANi, the A/D conversion is not interrupted during the conversion regardless of the priority. In this case, on completion of the A/D conversion on channel ANi, A/D conversion is performed according to the priority on remaining channels (in this case, ANj and ANk) in which interrupt conversion requests are pending. Therefore, the priority on interrupt conversion determines which channel is to be converted for the next operation. When a single trigger source generates interrupt conversion requests on two channels or multiple trigger sources simultaneously generate interrupt conversion requests, A/D conversion is performed according to this priority.

When interrupt conversion is requested during scan conversion, the scan conversion on channel ANi is suspended, and A/D conversion on the other channel (ANj) in which the interrupt conversion was requested is performed. On completion of the interrupt conversion on channel ANj, the scan conversion is resumed from the interrupted channel (ANi). This scheme ensures that the length of time required from the initiation of an interrupt conversion request to the completion of it is always constant. This makes it possible, for example, to perform A/D conversion in pin-point accuracy by synchronizing them with the operation of A/D conversion sources that are external to the LSI.

When the ADTF bit is set to 1 while the ADIDE bit in the ADTRD register is set to 1, an ADID interrupt is requested. To clear the ADTF bit to 0, write 0 to the ADTF bit after reading it as 1. If the DMAC or A-DMAC is started by an ADID interrupt, note that the ADTF bit is automatically cleared to 0 and the ADID interrupt is also cleared. The DMA transfer of the DMAC is supported on channels AN40 (ADID40) to AN47 (ADID47); and the DMA transfer of the A-DMAC is supported on channel AN0 (ADID0).

20.5.5 Example Operation of Interrupt Conversion

The following is an example operation of interrupt conversion where timer G0 is selected as a trigger source for channel AN40 and timer G2 is selected as a trigger source for channels AN42 and AN46.

1. Set bits ADTRGE40, ADTRGE42, and ADTRGE46 in A/D interrupt trigger enable register 1 (ADTRE1) to 1.
2. Clear bits ADTRS40, ADTRS42, and ADTRS46 in A/D interrupt trigger source select register 1 (ADTRS1) to 0.
3. Subsequently, interrupt conversion requests are generated by timers G0 and G2 at intervals specified by the ATU-III registers. For details on the ATU-III registers, see section 13, Advanced Timer Unit III (ATU-III).
4. When interrupt conversion is requested by timer G0, the ADITACT bit is set to 1 and interrupt conversion on channel AN40 is performed. On completion of A/D conversion on AN40, the A/D converted value of AN40 is transferred to ADR40 and bit ADTF40 in ADTRF1 is set to 1. The ADITACT bit is cleared to 0 and the interrupt conversion is completed. Furthermore, if bit ADIDE40 is 1, an ADID40 interrupt is requested to the CPU.
5. When interrupt conversion is requested by timer G2, the ADITACT bit is set to 1, and interrupt conversion on channels AN42 and AN46 is performed. Then A/D conversion on channel AN42 is performed. On completion of the conversion, the A/D converted value of AN42 is transferred to ADR42, and bits ADTF42 in ADTRF1 is set to 1. An A/D conversion on channel AN46 is then performed. On completion of the conversion, the A/D converted value of AN46 is transferred to ADR46, and bit ADTF46 in ADTRF1 is set to 1. The ADITACT bit is cleared to 0 and the interrupt conversion is completed. Further, if bits ADIDE42 and ADIDE46 in ADTRD1 are set to1 when either bit ADTF42 or ADTF46 is set to 1, the A/D converter requests an ADID42 or an ADID46 interrupt to the CPU.
6. Subsequently, steps 4 to 5 are repeated. The following is an example operation when requests by timers G0 and G2 conflict.

(1) Example Operation 1

When a timer G0 interrupt conversion request is input during the A/D conversion on channel AN42 due to a timer G2 interrupt conversion request, the request is processed as follows.

The timer G0 interrupt source is retained in the A/D converter until the conversion on channel AN42 is completed. On completion of the A/D conversion on channel AN42, A/D conversion on channels AN40 and AN46 is performed in the order according to the priority.

(2) Example Operation 2

When interrupt conversion requests by timers G0 and G2 are input simultaneously, the requests are processed as follows.

The timer G0 and timer G2 interrupt sources are retained in the A/D converter. A/D conversion on channels AN40, AN42, and AN46 in the order according to the priority.

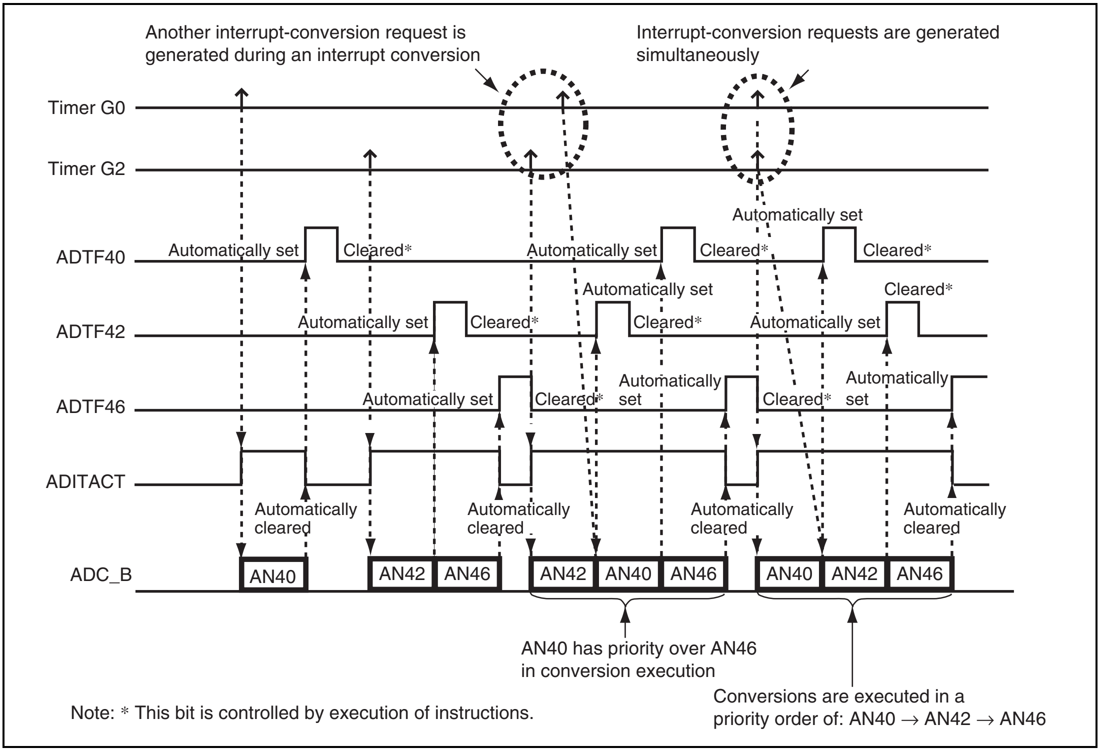

Another interrupt-conversion request is generated during an interrupt conversion
Interrupt-conversion requests are generated simultaneously
Timer G0
Timer G2
ADTF40
ADTF42
ADTF46
ADITACT
ADC_B
Automatically set
Cleared*
Automatically cleared
Automatically set
AN40
AN42
AN46
AN42
AN40
AN46
AN40
AN42
AN46
AN40 has priority over AN46 in conversion execution
Conversions are executed in a priority order of: AN40 → AN42 → AN46
Note: * This bit is controlled by execution of instructions.

Figure 20.6 Example Operation of Interrupt Conversion

20.5.6 Interrupt Conversion during Scan Conversion

The following is an example operation where single scan conversion on three channels AN0, AN3, and AN9 is started by a scan conversion request from timer G4 and then an interrupt conversion on channel AN6 is started by a interrupt conversion request from timer D03A.

1. Clear the ADCS and EXTRG bits in A/D control register 0 (ADCSR0) to 0, and set the TRGE bit in ADCSR0 to 1.
2. Set bits ANS0, ANS3 and ANS9 bits in A/D channel select register 0 (ADANS0) to 1.
3. Set bit ADTRGE6 in A/D interrupt trigger enable register 0 (ADTRE0) to 1.
4. Clear bit ADTRS6 in A/D interrupt trigger source select register (ADTRS0) to 0.
5. Subsequently, an scan conversion request is generated by timer G4 and an interrupt conversion request is generated by timer D03A at intervals specified by the ATU-III registers. For details on the ATU-III registers, see section 13, Advanced Timer Unit III (ATU-III).
6. When scan conversion is requested by timer G4, the ADSCACT bit is set to 1. Then, A/D conversion on channels AN0, AN3, and AN9 is performed in the order. On completion of the conversion, the ADF bit is set to 1 and the ADSCACT bit is cleared to 0, indicating that the scan conversion is completed.
7. When interrupt conversion is requested by timer D03A, the ADITACT bit is set to 1 and interrupt conversion on channel AN6 is performed. On completion of the A/D conversion on channel AN6, bit ADTF6 in ADTRF0 is set to 1 and the ADITACT bit is cleared to 0, indicating that the interrupt conversion is completed.
8. Subsequently, steps 6 to 7 are repeated. The following is an example operation where a scan conversion and an interrupt conversion conflict.

(1) Example Operation

When a timer D03A interrupt conversion request is input during the A/D conversion on channel AN3 in the scan conversion due to a timer G4 scan conversion request, the request is processed as follows.

The timer D03A interrupt source is retained in the A/D converter, and the scan conversion on channel AN3 is suspended. The priority is applied to channels AN3 and AN9 on which scan conversion is pending, and is applied to AN6 which is the current request. In this case, the A/D conversion on channels AN6, AN3, and AN9 in the order.

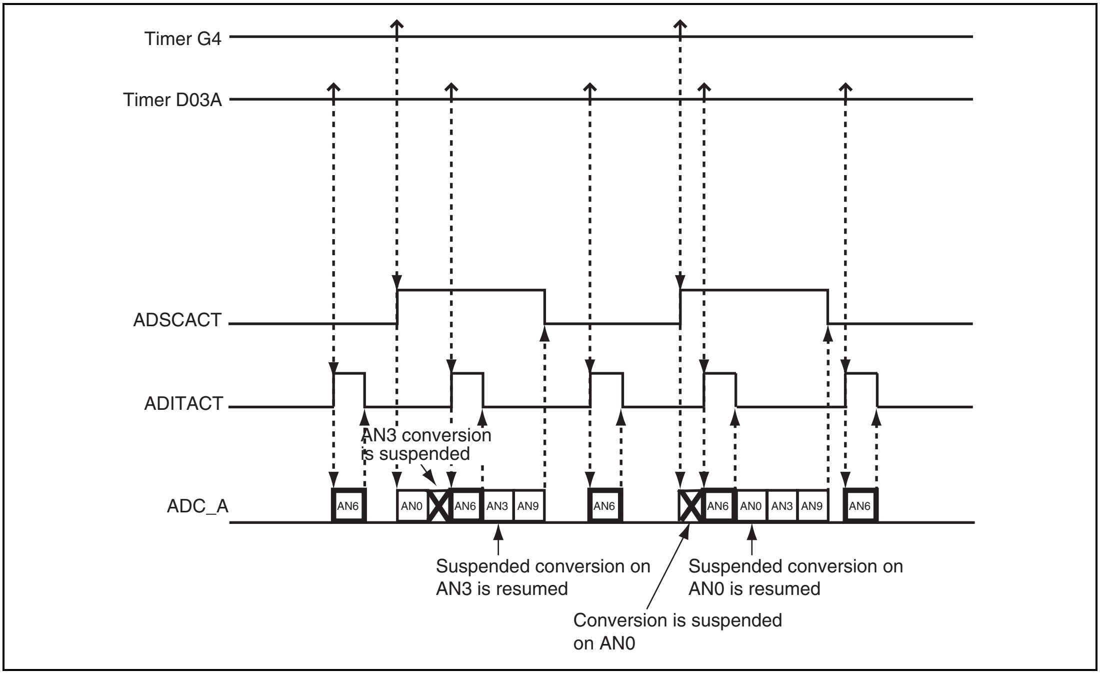

Timer G4
Timer D03A
ADSCACT
ADITACT
AN3 conversion is suspended
ADC_A
AN6
AN0
AN6
AN3
AN9
AN6
AN6
AN0
AN3
AN9
AN6
Suspended conversion on AN3 is resumed
Suspended conversion on AN0 is resumed
Conversion is suspended on AN0

Figure 20.7 Operation Example of Interrupt Conversion during Scan Conversion

20.5.7 Analog Input Sampling and Scan Conversion Time

The A/D converter includes sample and hold circuit. When start-of-scan-conversion delay time (t_p) have passed after the ADST bit in ADCSR is set to 1, the A/D converter samples the analog input, and then begins the conversion process.

Figure 20.8 shows a timing chart for a scan conversion on one channel in single-cycle scan mode. Scan conversion time (t_{SCAN}) includes start-of-scan-conversion delay time (t_p), analog input sampling time (t_{SPL}), A/D conversion processing time (t_{CONV}) and end-of-scan-conversion delay time (t_{ED}). The scan conversion time is shown in table 20.5.

The scan conversion time (t_{SCAN}) in single-cycle scan mode for which the number of selected channels is n can be determined according to the following equation:

$$t_{SCAN} = t_D + \{(t_{SPL} + t_{CONV}) \times n\} + t_{ED}$$

The scan conversion time for the first cycle in continuous scan mode is t_{SCAN} for single-cycle scan minus t_{ED}.

The scan conversion time for the second and subsequent cycles in continuous scan mode is a fixed time, which is equal to $\{(t_{SPL} + t_{CONV}) \times n\}$.

Table 20.5 Scan Conversion Time

		Based on Pϕ		
Item	**Symbol**	**Low Speed (CKS = 0)**	**High Speed (CKS = 1)**	**Unit**
Start-of-scan-conversion delay time	t_D	7	5	State
Analog input sampling time	t_{SPL}	20	10	
A/D conversion processing time	t_{CONV}	30	15	
End-of-scan-conversion delay time	t_{ED}	4	2	
Scan conversion time	t_{SCAN}	61	32	

Note: CKS cannot be set to 1 when two-time multiplication peripheral clock is set.

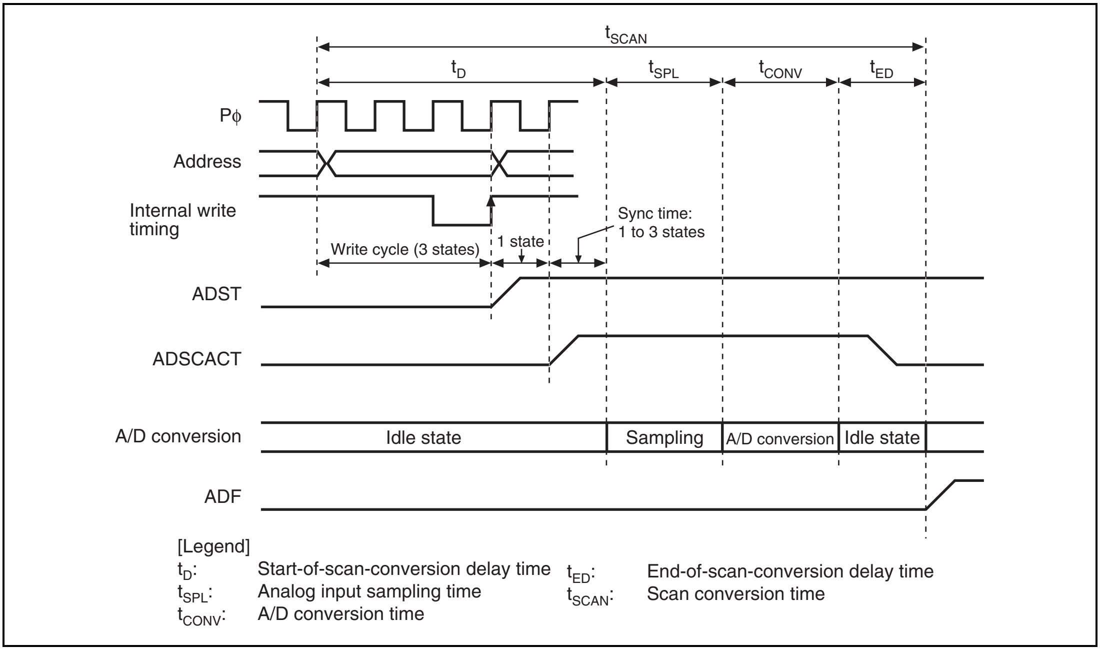


Figure 20.8 Timing Diagram for Scan Conversion (Single Channel, Single Cycle)

20.5.8 Starting Scan Conversion with External Trigger

The A/D converter can be activated by the input of an external trigger. To start up the A/D converter by an external trigger, the pin function should be set up using the pin function controller (PFC). After applying a high-level signal to the ADTRG pin, both the TRGE and EXTRG bits in the A/D control register (ADCSR) should be set to 1. If a low-level signal is then input to the ADTRG pin, the A/D converter detects a pulse fall edge and sets the ADSCSCT bit to 1.

Figure 20.9 shows an external of trigger input timing. If the low-speed is selected (CKS = 0), the timing at which the ADSCACT bit is set to 1 is 4 states after the falling edge of the ADTRG pin is sampled; while if the high-speed is selected (CKS = 1), it is 3 states after the fall edge of the ADTRG pin is sampled.

The timing at which a scan conversion is started after the ADSCACT bit is set to 1 is the same as the case where the ADST bit is set to 1 from 0 by software. For details on pin function setting, see section 23, Pin Function Controller (PFC).

To stop the scan conversion process while it is in progress, write 1 to the ADST bit and then write 0 to it.

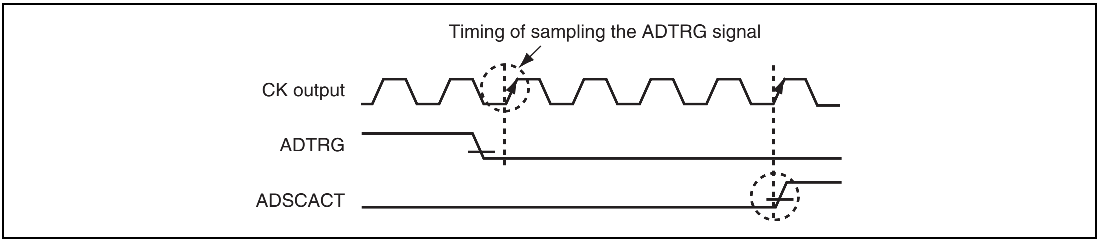


Figure 20.9 External Trigger Input Timing

20.5.9 Starting Scan Conversion with ATU-III Timer Trigger

A scan conversion can be activated by an ATU-III timer trigger. To start up a scan conversion by an ATU-III timer trigger, both the TRGE and EXTRG bits in the A/D control register (ADCSR) should be cleared to 0. If a timer trigger (timerG4 or timer G5) is entered in this situation, the ADSDACT bit is set to 1. The timing at which a scan conversion is started after the ADSCACT bit is set to 1 is the same as the case where the ADST bit is set to 1 from 0 by software.

To stop the scan conversion process while it is in progress, write 1 to the ADST bit and then write 0 to it.

20.5.10 Monitoring via ADEND_A and ADEND_B Output Pins

The timing at which AN0 and AN40 are scan-converted can be monitored via the ADEND_A and ADEND_B output pins, respectively. For details on pin function setting, see section 23, Pin Function Controller (PFC).

Figure 20.10 shows ADEND_A and ADEND_B output examples. If ADEND_A and ADEND_B outputs are selected by the PFC, monitor signals are output, respectively, from the ADEND_A and ADEND_B output pins during the conversion processing of channels AN0 and AN40. Upon completion of the AN0 and AN40 samplings, outputs are produced from the ADEND_A and ADEND_B pins, respectively.

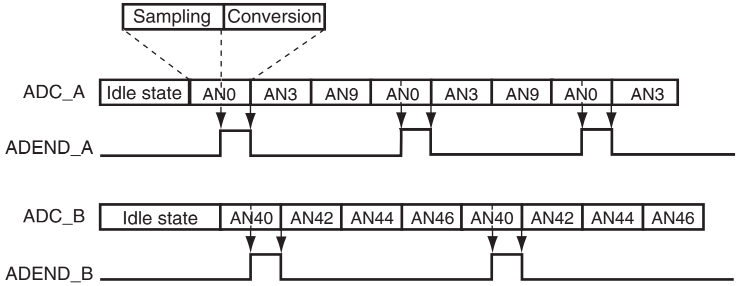


Figure 20.10 Example of ADEND_A and ADEND_B Outputs

Note: If an interrupt conversion is performed while high-level signals are output from the ADEND_A and ADEND_B pins, low-level signals are output once from these pins. After that, because channels AN0 and AN40 are converted again in a scan conversion process, high-level signals are output from the ADEND_A and ADEND_B pins again. In addition, if channels AN0 and AN40 are converted in an interrupt conversion process, high-level signals are also output from the ADEND_A and ADEND_B pins. Further, if channels AN0 and AN40 are set to A/D-converted value addition mode, high-level signals are output from the ADEND_A and ADEND_B pins only during the final A/D conversion (in the 4th conversion if four-addition conversion is performed, for example).

20.6 Interrupt Sources and DMA Transfer Request

20.6.1 Interrupt Requests on Completion of Scan Conversion

The A/D converter can generate a scan conversion end interrupt request (ADI) to the CPU. By setting the ADIE bit in the A/D control register (ADCSR) to 1, an ADI interrupt is enabled; by clearing the bit to 0, an ADI interrupt is disabled. In addition, the DMAC can be started up when an ADI interrupt is generated. In this case, interrupts are not generated to the CPU. If the DMAC is started upon an ADI interrupt, the ADF bit in the A/D conversion status register (ADREF) is automatically cleared to 0 when data transfer is performed by the DMAC.

For details on DMAC settings, see section 11, Direct Memory Access Controller (DMAC).

Note: The ADF bit is not cleared by an interrupt request to the CPU.

20.6.2 Interrupt Requests on Completion of Interrupt Conversion

The A/D converter can generate interrupt conversion end interrupt requests (ADID0 to ADID15, ADID40 to ADID47) to the CPU upon completion of an interrupt conversion. By setting the ADIDE0 to ADIDE15 and ADIDE40 to ADIDE47 bits in the A/D interrupt trigger processing end interrupt enable register (ADTRD) to 1, the ADID0 to ADID15 and ADID40 to ADID47 interrupts are enabled, respectively; by clearing the bits to 0, the ADID0 to ADID15 and ADID40 to ADID47 interrupts are disabled, respectively. If the DMAC is activated by an ADID40 to ADID47 interrupt, a corresponding bit in the ADTF40 to ADTF47 bits in the A/D interrupt trigger processing end flag register 1 (ADTRF1) is automatically cleared to 0 when data transfer is performed by the DMAC. If the A-DMAC is activated by an ADID0 interrupt, the ADTF0 bit in the A/D interrupt trigger processing end flag register 0 (ADTRF0) is automatically cleared to 0 when data transfer is performed by the ADMAC.

For details on DMAC settings, see section 11, Direct Memory Access Controller (DMAC) and section 12, Automotive Direct Memory Access Controller (A-DMAC).

Note: The ADTF bit is not cleared by an interrupt request to the CPU.

20.7 Definition of A/D Conversion Accuracy

The definition of A/D conversion accuracy is described below.

- Resolution

  This indicates the number of digital output codes in the A/D converter
- Quantization error

  This error, which is inherent to the A/D converter, is given as 1/2LSB (figure 20.11).
- Offset error

  This error, which is exclusive of quantization error, is a deviation of the analog input voltage value from the ideal A/D conversion characteristics when the digital output changes from a minimum voltage value B'000000000000 to B'000000000001 (figure 20.11).
- Full scale error

  This error, which is exclusive of quantization error, is a deviation of the analog input voltage value from the ideal A/D conversion characteristics when the digital output changes from B'111111111110 to B'111111111111 (figure 20.11).
- Nonlinearity error

  This error, which is exclusive of offset error, full scale error and quantization error, is a deviation from the ideal A/D conversion characteristics through the zero-scale and full-scale transitions (figure 20.11).
- Absolute accuracy

  This is a deviation of the digital value from the analog input value. This includes offset error, full scale error, quantization error, and nonlinearity error.

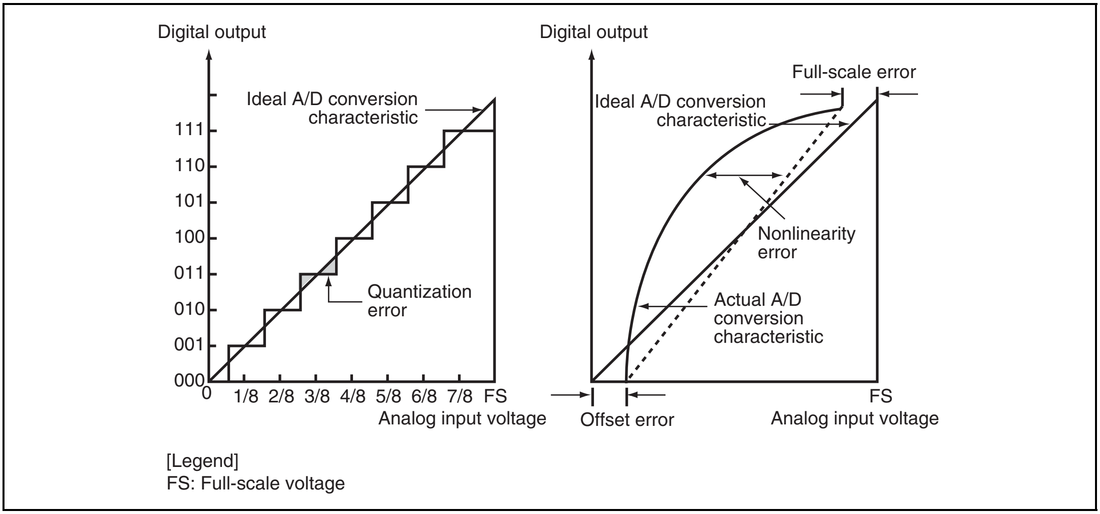


Figure 20.11 Definition of A/D Conversion Accuracy

20.8 Usage Notes

20.8.1 Analog Input Voltage Range

The voltage applied to an analog input pin (ANn) during A/D conversion should be within the following range:

AVrefl_A ≤ ANn (n = 0 to 27, 40 to 48) ≤ AVrefh_A

AVrefl_B ≤ ANn (n = 0 to 27, 40 to 48) ≤ AVrefh_B

20.8.2 Relationship among AV_{CC}, AV_{SS}, V_{CC}, and V_{SS}

When using the A/D converter, make sure that the following relationships are held among AV_{cc}, AV_{ss}, V_{cc} and V_{ss}:

AV_{cc} = 5.0 V ± 0.5 V, $AV_{ss} = V_{ss}$

When the A/D converter is not used, AV_{cc} pin must not be open. In this case, the following relationship should be held between AV_{ss} and V_{ss}:

$AV_{ss} = V_{ss}$

20.8.3 Allowable Settings for Pins AVrefh_A/AVrefh_B and AVrefl_A/AVrefl_B

The allowable settings for the AVrefh_A/AVrefh_B pins are as follows:

When the A/D converter is used: AVrefh_A = 4.5V to AV_{cc}, AVrefh_B = 4.5V to AV_{cc}

When the A/D converter is not used: AVrefh_A ≤ AV_{cc}, AVrefh_B ≤ AV_{cc}

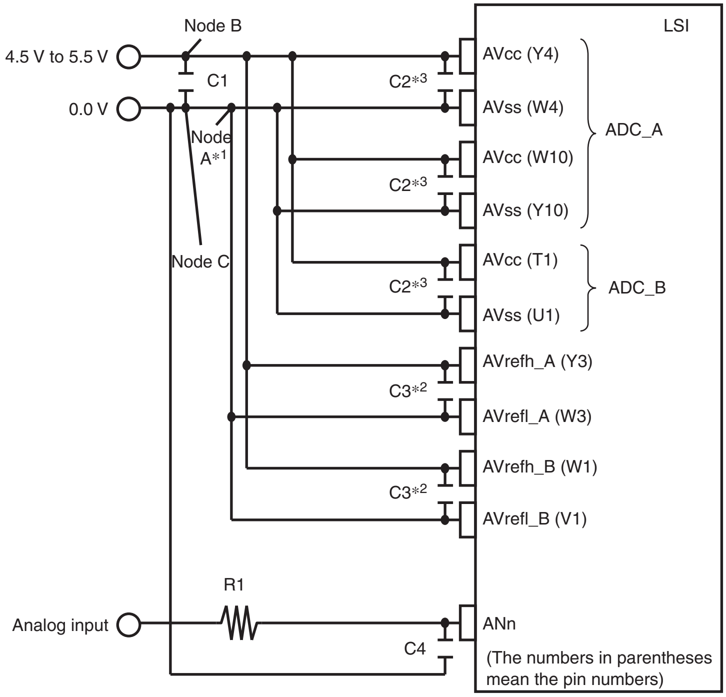
If any value outside the above range is set, it can adversely affect the reliability of the LSI. For the AVrefl_A/AVrefl_B pin, set AVrefl_A/AVrefl_B = $AV_{ss} = V_{ss}$.

20.8.4 Precautions on Board Design

For designing a board, to the maximum extent possible the digital circuits should be laid out separately from the analog circuits. Layouts involving the crossing of signal lines for digital circuits and signal lines for analog circuits, or placing them in proximity to each other, should be avoided. If the dissimilar signal lines are placed in close proximity to each other, the resulting induction can lead to a malfunction of the analog circuits or produce an adverse impact on A/D conversion values.

It should be noted that the analog input pins (AN0 to AN27, AN40 to AN48), the analog reference voltages (AVrefh_A/AVrefh_B, AVrefl_A/AVrefl_B), and the analog power supply (AV_{CC}) should be isolated from the digital circuits by means of analog grounding (AV_{SS}). In addition, the analog ground (AV_{SS}) should be connected in one point to a stable digital ground (V_{SS}) on the board.

As the protection circuit to prevent the analog input pins (AN0 to AN27, AN40 to AN48) from damages, by such abnormal voltages as excessive surges, bypass capacitors should be connected between AV_{CC} and AV_{SS}, and also between AVrefh_A/B and AVrefl_A/B, as illustrated in figures 20.12 (1) to 20.12 (3). Also a filter capacitor connected to an analog input pin (ANn) should be connected to the AV_{SS}. The capacitance values of the bypass capacitors connected between AV_{CC} and AV_{SS} or between AVrefh_A/B and AVrefl_A/B, as illustrated in figures 20.12 (1) to 20.12 (3), are reference values. Therefore, for designing a board, care must be taken to choose appropriate capacitance values. In addition, connecting a filter capacitor, as illustrated in figures 20.12 (1) to 20.12 (3), can cause an error by averaging the input currents to analog input pins (ANn). Therefore, care must be taken to choose appropriate circuit constants. Figure 20.12 (1) shows an example of connecting power supples in a basic configuration. Figure 20.12 (2) shows an example of connecting power supples in a configuration that is less affected by the board. Figure 20.12 (3) shows an example of connecting power supples in a configuration in which the voltages of AV_{CC} and AVrefh can differ from each other.

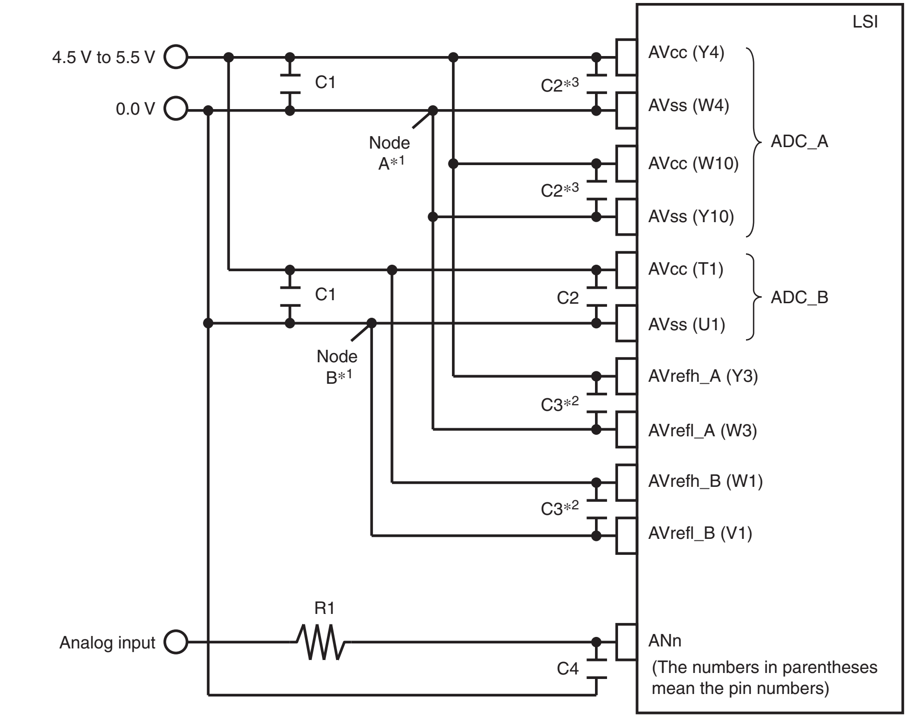


Reference values:

C1	: 10.0 µF to 100.0 µF (47 µF*5)
C2, C3	: 0.1 µF to 1.0 µF (0.1 µF*5)
C4	: 0.1 µF
R1	: 3 kΩ

Notes:
1. For designing a board, to reduce the potential difference between AVss and AVrefl, that may cause an offset error, the all wiring impedances between node A and AVss (W4), between node A and AVss (Y10), and between node A and AVrefl_A, must be the same value that is as low as possible. Similarly, the both wiring impedances between node A and AVss (U1) and between node A and AVrefl_B must be the same value that is as low as possible.
2. If an error is getting worse toward the full-scale voltage, the insufficient stability of the voltage of AVrefh_A/B may cause the problem.
   To improve the stability of the voltage, either reduce the wiring impedance of AVrefh_A/B, or increase the capacitance value of C3.
3. It is recommended that the C2 capacitors are connected between the respective pairs of AVcc and AVss. However, when the characteristics can be satisfied by the board design, with concerning the condition described in note 1, one C2 capacitor can be connected in common.
4. If an error is getting worse by the simultaneous operation of ADC_A and ADC_B, the mutual interference between ADC_A and ADC_B must be reduced. Thus, to reduce it, set the connecting node of AVcc (Y4, W10), AVcc (T1), AVrefh_A and AVrefh_B, closer to node B. Similarly, set the connecting node of AVss (W4, Y10), AVss (U1), AVrefl_A and AVrefl_B, closer to node C. In addition, increasing the capacitance value of C1 is also effective.
5. The capacitance values have been evaluated on a Renesas evaluation board. The most appropriate capacitance values should be evaluated on the actual board of the user.

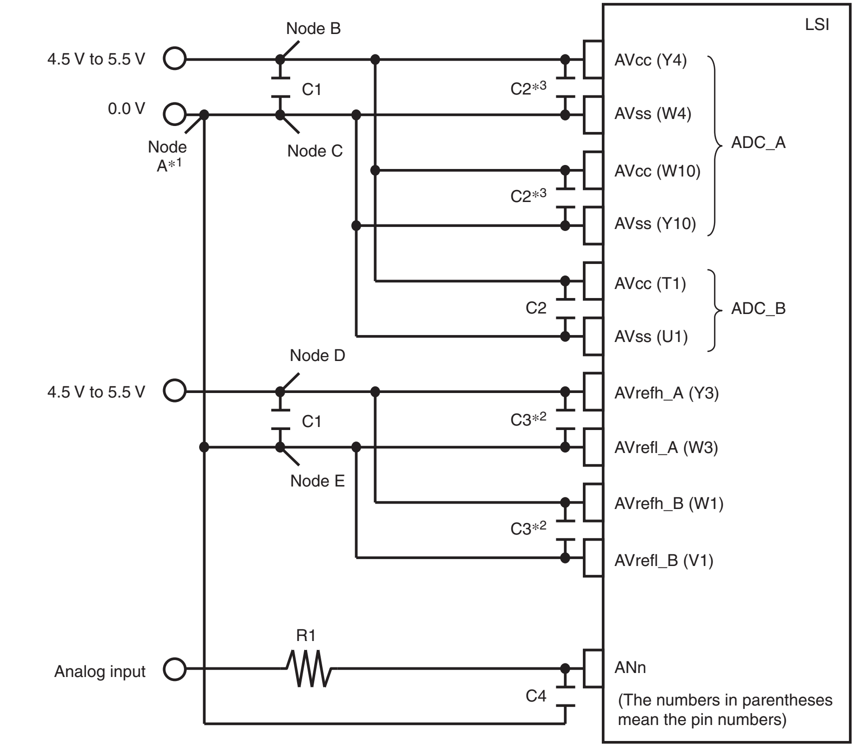
Figure 20.12 (1) Example of Connecting Analog Power Supples and Analog Input Pins (Basic Configuration)



Reference values: C1 : 10.0 µF to 100.0 µF (47 µF*5)
C2, C3 : 0.1 µF to 1.0 µF (0.1 µF*5)
C4 : 0.1 µF
R1 : 3 kΩ

Notes: 1. For designing a board, to reduce the potential difference between AVss and AVrefl, that may cause an offset error, the all wiring impedances between node A and AVss (W4), between node A and AVss (Y10), and between node A and AVrefl_A, must be the same value that is as low as possible. Similarly, the both wiring impedances between node B and AVss (U1) and between node B and AVrefl_B must be the same value that is as low as possible.
2. If an error is getting worse toward the full-scale voltage, the insufficient stability of the voltage of AVrefh_A/B may cause the problem.
To improve the stability of the voltage, either reduce the wiring impedance of AVrefh_A/B, or increase the capacitance value of C3.
3. It is recommended that the C2 capacitors for ADC_A are connected between the respective pairs of AVcc and AVss. However, when the characteristics can be satisfied by the board design, with concerning the condition described in note 1, one C2 capacitor can be connected in common.
4. If an error is getting worse by the simultaneous operation of ADC_A and ADC_B, increase the capacitance value of C1 to reduce the mutual interference between ADC_A and ADC_B.
5. The capacitance values have been evaluated on a Renesas evaluation board. The most appropriate capacitance values should be evaluated on the actual board of the user.

Figure 20.12 (2) Example of Connecting Analog Power Supples and Analog Input Pins (Less Affected by Board)



Reference values:		
	C1	: 10.0 µF to 100.0 µF (47 µF*6)
	C2, C3	: 0.1 µF to 1.0 µF (0.1 µF*6)
	C4	: 0.1 µF
	R1	: 3 kΩ

Notes: 1. For designing a board, to reduce the potential difference between AVss and AVrefl, that may cause an offset error, the all wiring impedances between node A and AVss (W4), between node A and AVss (Y10), and between node A and AVrefl_A, must be the same value that is as low as possible. Similarly, the both wiring impedances between node A and AVss (U1) and between node A and AVrefl_B must be the same value that is as low as possible.

2. If an error is getting worse toward the full-scale voltage, the insufficient stability of the voltage of AVrefh_A/B may cause the problem.
To improve the stability of the voltage, either reduce the wiring impedance of AVrefh_A/B, or increase the capacitance value of C3.

3. It is recommended that the C2 capacitors for ADC_A are connected between the respective pairs of AVcc and AVss. However, when the characteristics can be satisfied by the board design, with concerning the condition described in note 1, one C2 capacitor can be connected in common.

4. If an error is getting worse by the simultaneous operation of ADC_A and ADC_B, the mutual interference between ADC_A and ADC_B must be reduced. Thus, to reduce it, set the connecting node of AVcc (Y4, W10) and AVcc (T1) closer to node B. Similarly, set the connecting node of AVss (W4, Y10) and AVss (U1) closer to node C, the node of AVrefh_A and AVrefh_B to node D, the node of AVrefl_A and AVrefl_B to node E, respectively. In addition, increasing the capacitance value of C1 is also effective.

5. By separating the power supplies to AVcc and AVrefh from each other, the voltages of AVcc and AVrefh_A/B can differ from each other.
However, for designing a board, AVrefh_A/B = 4.5V to AVcc must be satisfied and the ground must be in common, with concerning the condition described in note 1.

6. The capacitance values have been evaluated on a Renesas evaluation board. The most appropriate capacitance values should be evaluated on the actual board of the user.

Figure 20.12 (3) Example of Connecting Analog Power Supples and Analog Input Pins (AVcc and AVrefh_A/B Can Differ from Each Other)

Section 21 JTAG Interface

This LSI includes the JTAG interface, providing the boundary scan function conforming to the IEEE standard 1149.1.

21.1 Features

- Five test signals: TCK, TDI, TDO, TMS, and $\overline{\text{TRST}}$
- TAP controller
- Four registers:

  Instruction register (SDIR), ID register (SDID), bypass register (SDBPR), and boundary scan register (SDBSR)
- Six commands conforming to the IEEE standard 1419.1:

  BYPASS, EXTEST, SAMPLE/PRELOAD, CLAMP, HIGHZ, and IDCODE

Figure 21.1 shows a block diagram of the JTAG interface.

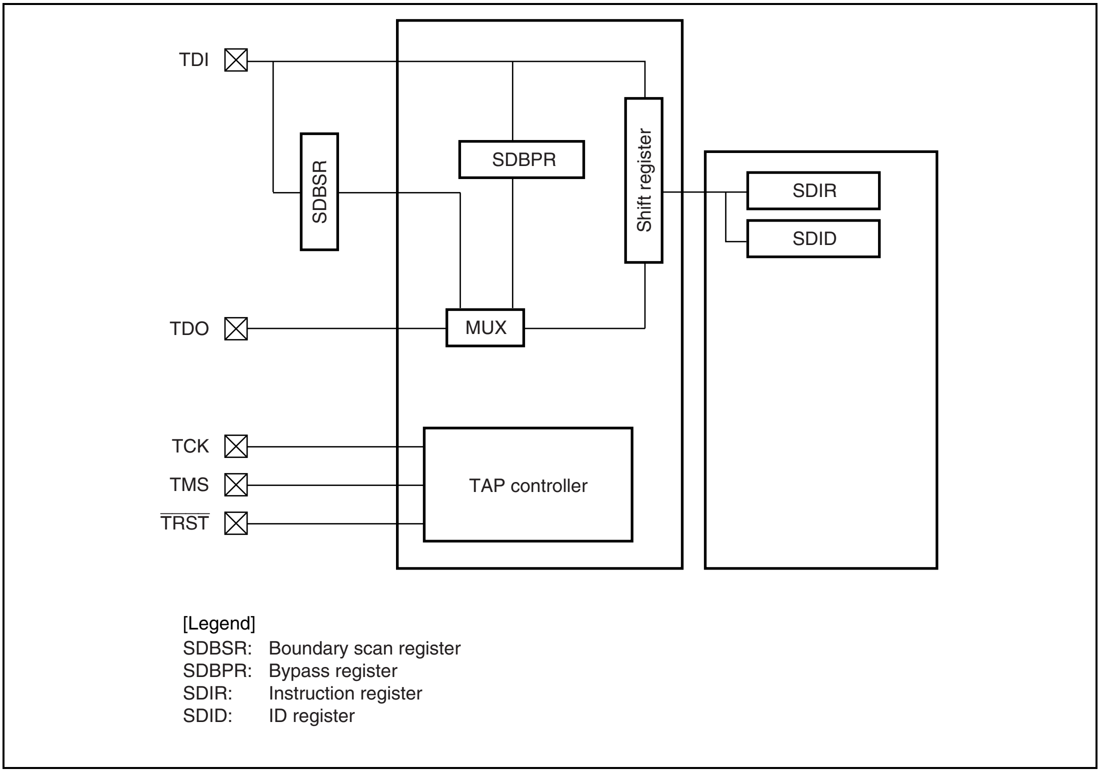


Figure 21.1 Block Diagram of JTAG Interface

21.2 Input/Output Pins

Table 21.1 shows the pin configuration of the JTAG interface.

Table 21.1 Pin Configuration

Pin Name	Description
TCK	Serial data input/output clock pin Data is supplied via the data input pin (TDI) and output from the data output pin (TDO) synchronously with the serial data input/output clock signal.
TMS	Mode select input pin The state of the TAP controller is determined by changing the level of this signal synchronously with the TCK signal. For the protocol, see figure 21.2.
$\overline{\text{TRST}}$	Reset input pin The input signal to this pin is accepted asynchronously with the TCK signal; a low-level input signal resets the JTAG interface. This pin must be held low for a specified time after turning on the power whether the JTAG interface is used or not.
TDI	Serial data input pin Data is sent to the JTAG interface by changing the level of this pin synchronously with the TCK signal.
TDO	Serial data output pin Data is read out from the JTAG interface by reading the level of this pin synchronously with the TCK signal.

21.3 Register Descriptions

The JTAG interface has the following registers. No register can be accessed by the CPU.

Table 21.2 Register Configuration

Register Name	Abbreviation	Initial Value*1	Size
Instruction register	SDIR	H'4	4
ID register	SDID	H'08086447	32
Bypass register	SDBPR	Undefined*2	1
Boundary scan register	SDBSR	Undefined	—

Note: *1 The registers are initialized while the TAP controller is in the Test-Logic-Reset state.
*2 Cleared in the Capture-DR state of the BYPASS instruction.

Commands and data can be input to SDIR and SDDR via the serial data input pin (TDI) using serial transfers. Data can be output from SDIR and SDDR via the serial data output pin (TDO). SDBPR, which is a 1-bit register, is connected between the TDI and TDO pins in BYPASS, CLAMP, and HIGHZ modes. SDBSR, which is a 457-bit boundary scan register, is connected between the TDI and TDO pins in SAMPLE/PRELOAD and EXTEST modes. SDID, which is a 32-bit register, can output the fixed code via the TDO pin in IDCODE mode.

Table 21.3 shows the possible serial transfers for the JTAG interface registers.

Table 21.3 Serial Transfers for JTAG Interface Registers

Register Abbreviation	Serial Input	Serial Output
SDIR	Available	Not available*
SDBPR	Available	Available
SDBSR	Available	Available
SDID	Not available	Available

Notes: * A fixed value (B'1101) is read.

21.3.1 Instruction Register (SDIR)

SDIR is a 4-bit register that holds one of the boundary scan commands. SDIR is initialized when the $\overline{\text{TRST}}$ pin is asserted or the TAP controller is in the Test-Logic-Reset state. If any reserved command is set in SDIR, the BYPASS instruction will be executed.

Table 21.4 Boundary Scan Commands

Command Codes				Description
0	0	0	0	JTAG EXTEST
0	0	0	1	JTAG SAMPLE/PRELOAD
0	1	1	0	JTAG CLAMP
0	1	1	1	JTAG HIGHZ
0	1	0	0	JTAG IDCODE (initial value)
1	1	1	1	JTAG BYPASS
Other than the above				Reserved

21.3.2 ID Register (SDID)

SDID is a 32-bit register that holds the LSI's ID code. SDID can be read out via the JTAG interface pins when the IDCODE command is set and cannot be written.

The read value is H'08086447; however, the upper four bits may be changed as the LSI version is revised.

21.3.3 Bypass Register (SDBPR)

SDBPR is a 1-bit register and is connected between the TDI and TDO pins when SDIR is set to BYPASS mode. The value in SDBPR is not initialized by a power-on reset or $\overline{\text{TRST}}$ pin assertion, but the first value read by the BYPASS instruction is always 0 because this register is initialized in the Capture-DR state of this instruction.

21.3.4 Boundary Scan Register (SDBSR)

SDBSR is a shift register arranged on a pad to control the external input and output pins. SDBSR is connected between the TDI and TDO pins when SDIR is set to SAMPLE/PRELOAD or EXTEST mode. The value in SDBSR is not initialized even by a power-on reset or $\overline{\text{TRST}}$ pin assertion and stays undefined.

Table 21.5 shows the relationship between the pins of this LSI and the SDBSR bits.

Table 21.5 Relationship between Pins and SDBSR Bits

Pin No.	Pin Name	Input/Output	Bit No.
From TDI			
C15	NMI	Input	456
B16	MD_CLK1	Input	455
D14	MD_CLK0	Input	454
B15	MD_CLKP	Input	453
C13	MD3	Input	452
A11	FWE	Input	451
B11	$\overline{\text{WDTOVF}}$	Output enabled	450
		Output	449
C11	PD13/TCLKB/TIJ1/TOE60	Output enabled	448
		Output	447
		Input	446
D10	PD12/TCLKA/TIOC41/TIJ0	Output enabled	445
		Output	444
		Input	443
A9	PD11/TIOC23/TIF2B/TOE51	Output enabled	442
		Output	441
		Input	440
C10	PD10/TIOC22/TIF1B/TOE50	Output enabled	439
		Output	438
		Input	437

Pin No.	Pin Name	Input/Output	Bit No.
B9	PD9/TIOC21/TIF0B/TOE43	Output enabled	436
		Output	435
		Input	434
A8	PD8/TIOC20/TIOC33/TOE53	Output enabled	433
		Output	432
		Input	431
A7	PD7/TIOC13/TOE42	Output enabled	430
		Output	429
		Input	428
A6	PD6/TIOC12/TOE41	Output enabled	427
		Output	426
		Input	425
B8	PD5/TIOC11/TOE23/TOE40	Output enabled	424
		Output	423
		Input	422
C9	PD4/TIOC10/TIOC32/TOE52	Output enabled	421
		Output	420
		Input	419
A5	PD3/TIOC03/TOE22/TOE53	Output enabled	418
		Output	417
		Input	416
A4	PD2/TIOC02/TOE21/TOE52	Output enabled	415
		Output	414
		Input	413
B7	PD1/TIOC01/TOE20	Output enabled	412
		Output	411
		Input	410
B6	PD0/TIOC00/TIOC31	Output enabled	409
		Output	408
		Input	407

Pin No.	Pin Name	Input/Output	Bit No.
D8	PE13/TOE13	Output enabled	406
		Output	405
		Input	404
C8	PE12/TOE12	Output enabled	403
		Output	402
		Input	401
A3	PE11/TOE11	Output enabled	400
		Output	399
		Input	398
A2	PE10/TOE10	Output enabled	397
		Output	396
		Input	395
C7	PE9/TOE03	Output enabled	394
		Output	393
		Input	392
B5	PE8/TOE02	Output enabled	391
		Output	390
		Input	389
D6	PE7/TOE01/CRx_B	Output enabled	388
		Output	387
		Input	386
B4	PE6/TOE00/CTx_B	Output enabled	385
		Output	384
		Input	383
B3	PE5/TIA05/TOE63/TIF25	Output enabled	382
		Output	381
		Input	380
C4	PE4/TIA04/TOE62/TIF24	Output enabled	379
		Output	378
		Input	377

Pin No.	Pin Name	Input/Output	Bit No.
C5	PE3/TIA03/TOE61/TIF23	Output enabled	376
		Output	375
		Input	374
C6	PE2/TIA02/TIOC43/TIOC30	Output enabled	373
		Output	372
		Input	371
D5	PE1/TIA01/TIOC42/TIOC40	Output enabled	370
		Output	369
		Input	368
D4	PE0/TIA00	Output enabled	367
		Output	366
		Input	365
C3	PH5/TIF5	Output enabled	364
		Output	363
		Input	362
D3	PH4/TIF4	Output enabled	361
		Output	360
		Input	359
B2	PH3/TIF3	Output enabled	358
		Output	357
		Input	356
F4	PH2/TIF2A	Output enabled	355
		Output	354
		Input	353
E3	PH1/ADTRG_B/TIF1A	Output enabled	352
		Output	351
		Input	350
B1	PH0/ADTRG_A/TIF0A	Output enabled	349
		Output	348
		Input	347

Pin No.	Pin Name	Input/Output	Bit No.
C2	PK11/MISOC	Output enabled	346
		Output	345
		Input	344
D2	PK10/MOSIC	Output enabled	343
		Output	342
		Input	341
F3	PK9/RSPCKC	Output enabled	340
		Output	339
		Input	338
E2	PK8/RxD_E	Output enabled	337
		Output	336
		Input	335
C1	PK7/TxD_E	Output enabled	334
		Output	333
		Input	332
D1	PK6/SCK_E	Output enabled	331
		Output	330
		Input	329
F2	PK5/RxD_D/MISOB	Output enabled	328
		Output	327
		Input	326
G3	PK4/TxD_D/MOSIB	Output enabled	325
		Output	324
		Input	323
E1	PK3/SCK_D/RSPCKB	Output enabled	322
		Output	321
		Input	320
F1	PK2/RxD_C/MISOA	Output enabled	319
		Output	318
		Input	317

Pin No.	Pin Name	Input/Output	Bit No.
H4	PK1/TxD_C/MOSIA	Output enabled	316
		Output	315
		Input	314
G2	PK0/SCK_C/RSPCKA/$\overline{\text{UBCTRG}}$	Output enabled	313
		Output	312
		Input	311
G1	PG0/TOD00A/SSLA0	Output enabled	310
		Output	309
		Input	308
H3	PG1/TOD01A/SSLA1	Output enabled	307
		Output	306
		Input	305
H2	PG2/TOD02A/SSLA2	Output enabled	304
		Output	303
		Input	302
H1	PG3/TOD03A/SSLA3	Output enabled	301
		Output	300
		Input	299
J1	PG4/TOD10A/SSLA4/SSLB3	Output enabled	298
		Output	297
		Input	296
J3	PG5/TOD11A/SSLA5/SSLC3	Output enabled	295
		Output	294
		Input	293
J2	PG6/TOD12A/SSLB0	Output enabled	292
		Output	291
		Input	290
K4	PG7/TOD13A/SSLB1	Output enabled	289
		Output	288
		Input	287

Pin No.	Pin Name	Input/Output	Bit No.
L1	PG8/TOD20A/SSLB2/TIF6	Output enabled	286
		Output	285
		Input	284
K3	PG9/TOD21A/SSLC0/TIF7	Output enabled	283
		Output	282
		Input	281
L2	PG10/TOD22A/SSLC1/TIF8	Output enabled	280
		Output	279
		Input	278
L3	PG11/TOD23A/SSLC2/TIF9	Output enabled	277
		Output	276
		Input	275
M1	PG12/TOD30A/SSLA4/TIF10	Output enabled	274
		Output	273
		Input	272
M4	PG13/TOD31A/SSLA5/TIF11	Output enabled	271
		Output	270
		Input	269
M2	PG14/TOD32A/SSLA6/TIF12	Output enabled	268
		Output	267
		Input	266
N1	PG15/TOD33A/SSLA7/TIF13	Output enabled	265
		Output	264
		Input	263
P1	PL8/TOE33	Output enabled	262
		Output	261
		Input	260
M3	PL7/TOE32/$\overline{\text{IRQ7}}$	Output enabled	259
		Output	258
		Input	257

Pin No.	Pin Name	Input/Output	Bit No.
N2	PL6/TOE31/$\overline{\text{IRQ6}}$	Output enabled	256
		Output	255
		Input	254
R1	PL5/TOE30/$\overline{\text{IRQ5}}$	Output enabled	253
		Output	252
		Input	251
P2	PL4/TOE23/$\overline{\text{IRQ4}}$	Output enabled	250
		Output	249
		Input	248
R2	PL3/TOE22/$\overline{\text{IRQ3}}$	Output enabled	247
		Output	246
		Input	245
N3	PL2/TOE21/$\overline{\text{IRQ2}}$	Output enabled	244
		Output	243
		Input	242
P4	PL1/TOE20/$\overline{\text{IRQ1}}$/$\overline{\text{POD}}$	Output enabled	241
		Output	240
		Input	239
P3	PL0/$\overline{\text{IRQ0}}$	Output enabled	238
		Output	237
		Input	236
V11	PF0/TOD00B/TIF6	Output enabled	235
		Output	234
		Input	233
Y13	PF1/TOD01B/TIF7	Output enabled	232
		Output	231
		Input	230
Y14	PF2/TOD02B/TIF8	Output enabled	229
		Output	228
		Input	227

Pin No.	Pin Name	Input/Output	Bit No.
W13	PF3/TOD03B/TIF9	Output enabled	226
		Output	225
		Input	224
V12	PF4/TOD10B/TIF10	Output enabled	223
		Output	222
		Input	221
Y15	PF5/TOD11B/TIF11	Output enabled	220
		Output	219
		Input	218
Y16	PF6/TOD12B/TIF12	Output enabled	217
		Output	216
		Input	215
U13	PF7/TOD13B/TIF13	Output enabled	214
		Output	213
		Input	212
W14	PF8/TOD20B/TIF14	Output enabled	211
		Output	210
		Input	209
Y17	PF9/TOD21B/TIF15	Output enabled	208
		Output	207
		Input	206
V13	PF10/TOD22B/TIF16	Output enabled	205
		Output	204
		Input	203
Y18	PF11/TOD23B/TIF17	Output enabled	202
		Output	201
		Input	200
V14	PF12/TOD30B/TIF18	Output enabled	199
		Output	198
		Input	197

Pin No.	Pin Name	Input/Output	Bit No.
W15	PF13/TOD31B/TIF19	Output enabled	196
		Output	195
		Input	194
Y19	PF14/TOD32B/CTx_B/TxD_A/FRTxD_B	Output enabled	193
		Output	192
		Input	191
W16	PF15/TOD33B/CRx_B/RxD_A/FRRxD_B	Output enabled	190
		Output	189
		Input	188
W17	PJ9/RxD_B/CRx_D	Output enabled	187
		Output	186
		Input	185
W18	PJ8/TxD_B/CTx_D	Output enabled	184
		Output	183
		Input	182
U15	PJ7/SCK_B/ADEND_A/TIJ1/FREN_A	Output enabled	181
		Output	180
		Input	179
W19	PJ6/RxD_A	Output enabled	178
		Output	177
		Input	176
V15	PJ5/TxD_A	Output enabled	175
		Output	174
		Input	173
T17	PJ4/SCK_A/ADEND_B/TIJ0	Output enabled	172
		Output	171
		Input	170
U17	PJ3/RxD_A/CRx_C/CRx_A&CRx_B&CRx_C/RxD_B/CRx_C&CRx_D	Output enabled	169
		Output	168
		Input	167

Pin No.	Pin Name	Input/Output	Bit No.
V16	PJ2/TxD_A/CTx_C/CTx_A&CTx_B&CTx_C/TxD_B/CTx_C&CTx_D	Output enabled	166
		Output	165
		Input	164
V17	PJ1/RxD_A/CRx_A/CRx_A&CRx_B/RxD_B/FRRxD_A	Output enabled	163
		Output	162
		Input	161
V18	PJ0/TxD_A/CTx_A/CTx_A&CTx_B/TxD_B/FRTxD_A	Output enabled	160
		Output	159
		Input	158
T18	PC15/D15/TOD13A/SSLA3	Output enabled	157
		Output	156
		Input	155
U18	PC14/D14/TOD12A/SSLA2	Output enabled	154
		Output	153
		Input	152
V19	PC13/D13/TOD11A/SSLA1	Output enabled	151
		Output	150
		Input	149
W20	PC12/D12/TOD10A/SSLA0	Output enabled	148
		Output	147
		Input	146
V20	PC11/D11/TOD03A/SSLB3	Output enabled	145
		Output	144
		Input	143
R18	PC10/D10/TOD02A/SCK_A	Output enabled	142
		Output	141
		Input	140
U19	PC9/D9/TOD01A/RxD_A/CRx_A	Output enabled	139
		Output	138
		Input	137

Pin No.	Pin Name	Input/Output	Bit No.
U20	PC8/D8/TOD00A/TxD_A/CTx_A	Output enabled	136
		Output	135
		Input	134
P18	PC7/D7/ADEND_A	Output enabled	133
		Output	132
		Input	131
T19	PC6/D6/ADTRG_A	Output enabled	130
		Output	129
		Input	128
R19	PC5/D5	Output enabled	127
		Output	126
		Input	125
T20	PC4/D4	Output enabled	124
		Output	123
		Input	122
N17	PC3/D3	Output enabled	121
		Output	120
		Input	119
P19	PC2/D2	Output enabled	118
		Output	117
		Input	116
R20	PC1/D1	Output enabled	115
		Output	114
		Input	113
N18	PC0/D0	Output enabled	112
		Output	111
		Input	110
P20	PB14/RD/$\overline{\text{WR}}$	Output enabled	109
		Output	108
		Input	107

Pin No.	Pin Name	Input/Output	Bit No.
N19	PB13/$\overline{\text{CS3}}$/RSPCKB/TIF22	Output enabled	106
		Output	105
		Input	104
N20	PB12/$\overline{\text{CS2}}$/RSPCKA/FREN_B	Output enabled	103
		Output	102
		Input	101
M20	PB11/$\overline{\text{CS1}}$/TOE21/TIF21/SSLB2	Output enabled	100
		Output	99
		Input	98
M18	PB10/$\overline{\text{CS0}}$/SSLB1	Output enabled	97
		Output	96
		Input	95
M19	PB9/$\overline{\text{RD}}$/SSLB0	Output enabled	94
		Output	93
		Input	92
K20	PB8/$\overline{\text{WAIT}}$/TOE20/TIF20/RxD_A	Output enabled	91
		Output	90
		Input	89
L17	PB7/$\overline{\text{WE1}}$/TxD_A	Output enabled	88
		Output	87
		Input	86
K19	PB6/$\overline{\text{WE0}}$/SCK_B	Output enabled	85
		Output	84
		Input	83
J20	PB5/A21/CRx_B/TIF7/TIF27/RxD_B	Output enabled	82
		Output	81
		Input	80
H20	PB4/A20/CTx_B/TIF6/TIF26/TxD_B	Output enabled	79
		Output	78
		Input	77

Pin No.	Pin Name	Input/Output	Bit No.
L18	PB3/A19/MISOB/TIA05/SSLA7	Output enabled	76
		Output	75
		Input	74
J19	PB2/A18/MOSIB/TIA04/SSLA6	Output enabled	73
		Output	72
		Input	71
K18	PB1/A17/MISOA/FRRxD_B	Output enabled	70
		Output	69
		Input	68
G20	PB0/A16/MOSIA/FRTxD_B	Output enabled	67
		Output	66
		Input	65
F20	PA15/A15/TIA03	Output enabled	64
		Output	63
		Input	62
H19	PA14/A14/TIA02	Output enabled	61
		Output	60
		Input	59
E20	PA13/A13/TIA01	Output enabled	58
		Output	57
		Input	56
D20	PA12/A12/TIA00	Output enabled	55
		Output	54
		Input	53
G19	PA11/A11/TOD13B/TIF2B	Output enabled	52
		Output	51
		Input	50
J18	PA10/A10/TOD12B/TIF1B	Output enabled	49
		Output	48
		Input	47

Pin No.	Pin Name	Input/Output	Bit No.
C20	PA9/A9/TOD11B/TIF0B	Output enabled	46
		Output	45
		Input	44
H18	PA8/A8/TOD10B	Output enabled	43
		Output	42
		Input	41
B20	PA7/A7/TOD03B/TIF2A	Output enabled	40
		Output	39
		Input	38
F19	PA6/A6/TOD02B/TIF1A	Output enabled	37
		Output	36
		Input	35
G17	PA5/A5/TOD01B/TIF0A	Output enabled	34
		Output	33
		Input	32
D19	PA4/A4/TOD00B	Output enabled	31
		Output	30
		Input	29
E19	PA3/A3	Output enabled	28
		Output	27
		Input	26
C19	PA2/A2	Output enabled	25
		Output	24
		Input	23
B19	PA1/A1	Output enabled	22
		Output	21
		Input	20
C18	PA0/A0	Output enabled	19
		Output	18
		Input	17

Pin No.	Pin Name	Input/Output	Bit No.
D18	$\overline{\text{AUDSYNC}}$	Output	16
		Output enabled	15
		Input	14
E18	AUDATA0	Output	13
		Output enabled	12
		Input	11
E17	AUDATA1	Output	10
		Output enabled	9
		Input	8
E18	AUDATA2	Output	7
		Output enabled	6
		Input	5
D17	AUDATA3	Output	4
		Output enabled	3
		Input	2
C17	AUDMD	Input	1
D16	$\overline{\text{AUDRST}}$	Input	0
To TDO			

21.4 Operations

21.4.1 TAP Controller

Figure 21.2 shows the internal state transition of the TAP controller.

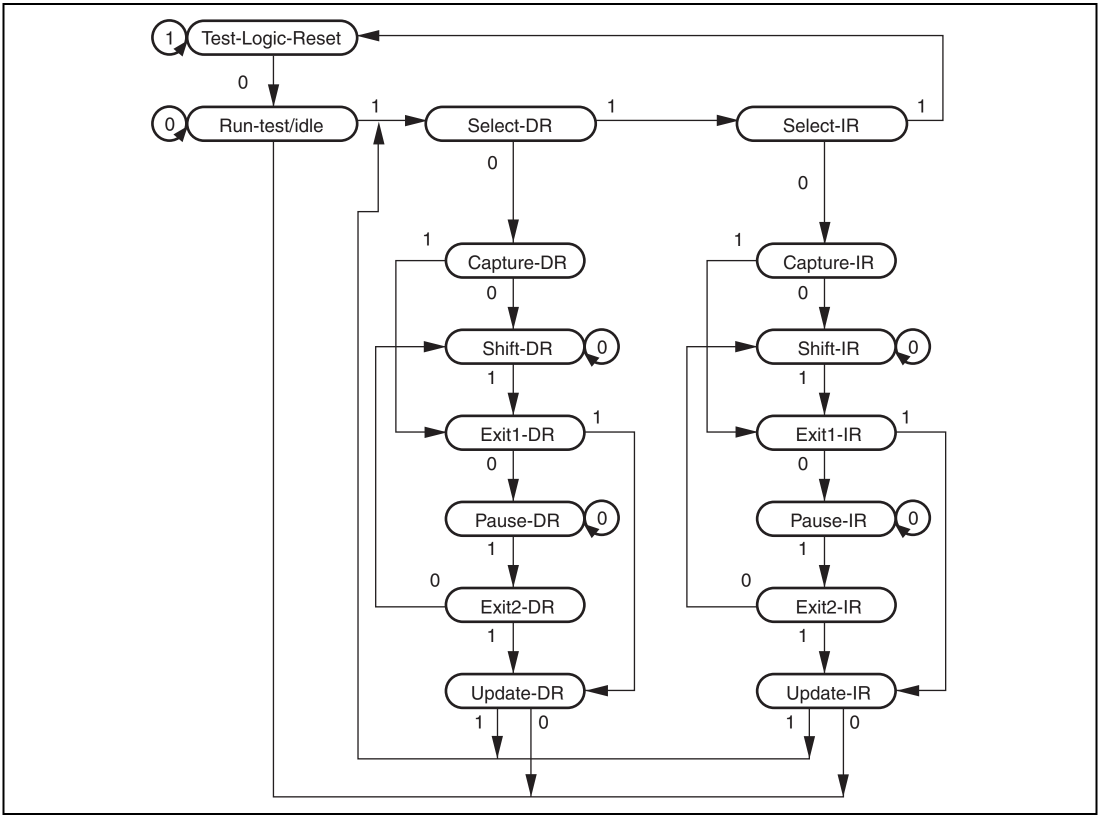


Figure 21.2 State Transition of TAP Controller

Note: A transition occurs depending on the TMS value at the rising edge of the TCK signal. The TDI value is sampled at the rising edge of the TCK signal and is shifted at the falling edge of the TCK signal. The TDO pin is always in the high-impedance state except in the Shift-DR and Shift-IR states. When the $\overline{\text{TRST}}$ pin is asserted, a transition to the Test-Logic-Reset state occurs asynchronously with the TCK signal.

21.4.2 Supported Commands

(1) BYPASS

The BYPASS command is an indispensable standard command for operating SDBPR. This command shortens the shift path thus transferring serial data for other LSIs on a printed-circuit board at a higher speed. While this command is being executed, the test circuit has no effect on the system circuit.

(2) SAMPLE/PRELOAD

The SAMPLE/PRELOAD command inputs data to SDBSR from the internal circuits of this LSI, outputs the data via the scan path, and loads data to the scan path. While this command is being executed, the values of the input pins of this LSI are sent directly to the internal circuits of this LSI, whereas the values of the internal circuits are output directly from the output pins. Executing this command has no effect on the system circuit.

The SAMPLE operation latches the snapshots of the values transferred from the input pins to the internal circuits and the values transferred from the internal circuits to the output pins and reads out the snapshots via the scan path. Here, the snapshots are latched without disturbing the operations of this LSI.

The PRELOAD operation sets the initial value to the parallel output latch circuit of the SDBSR via the scan path, prior to the EXTEST command. If the EXTEST command is executed without PRELOAD operation, undefined data is output from the output pins (with the EXTEST command, the values are always output to the output pins from the parallel output latch circuit) until the first scan sequence is complete (until data transfer to the output latch circuit is complete).

(3) EXTEST

The EXTEST command tests external circuits when this LSI is mounted on a printed-circuit board. When this command is executed, the output pins are used to output the test data, which has been set using the SAMPLE/PRELOAD command, to the printed-circuit board from the SDBSR, whereas the input pins are used to latch the test result to SDBSR from the printed-circuit board. When the EXTEST command is used N times for a test, the test data for the Nth command is scanned in when the test data for the (N – 1)th command is scanned out.

The data that has been loaded on SDBSR of the output pins in the Capture-DR state of this command is not used to test the external circuits (it is shifted and replaced).

(4) CLAMP

The CLAMP command allows the output pins to output the value from SDBSR, which has been set using the SAMPLE/PRELOAD command. While the CLAMP command is being selected, SDBSR retains the previous state regardless of the TAP controller state.

SDBPR is connected between the TDI and TDO pins, thus allowing the same operation as that when the BYPASS command is selected.

(5) HIGHZ

The HIGHZ command drives all the output pins to the high-impedance state. While the HIGHZ command is being selected, SDBSR retains the previous state regardless of the TAP controller state.

SDBPR is connected between the TDI and TDO pins, thus allowing the same operation as that when the BYPASS command is selected.

(6) IDCODE

The IDCODE command sets the JTAG interface pins to IDCODE mode prescribed in the JTAG. Specifically, the pins are set to IDCODE mode when the JTAG interface is initialized, that is, when the $\overline{\text{TRST}}$ pin is asserted or the TAP controller is placed in the Test-Logic-Reset state.

21.4.3 Notes

The JTAG interface of the LSI has the following restrictions.

- Clock-related signals (EXTAL and XTAL) cannot be boundary-scanned.
- Reset-related signals ($\overline{\text{RES}}$ and $\overline{\text{HSTBY}}$) cannot be boundary-scanned.
- JTAG interface-related signals (TCK, TDI, TDO, TMS, and $\overline{\text{TRST}}$) cannot be boundary-scanned.
- MD0, MD1, MD2, MD4, ASEMD, AUDCK, and CK cannot be boundary-scanned.
- When using the JTAG interface, pins ASEMD, MD0, MD1, MD2, and MD4 must be 0 and pins $\overline{\text{RES}}$ and $\overline{\text{HSTBY}}$ must be 1.
- ADC-related pins (AN0 to AN27 and AN40 to AN48) cannot be boundary-scanned.
- Fix the $\overline{\text{RES}}$ pin to the low level while the EXTEST, CLAMP, or HIGHZ command is being set.
- Selecting HIGHZ command for the following pins is invalid, as they are set to be pulled up/down.

  Always invalid: AUDMD, $\overline{\text{AUDSYNC}}$, AUDATA, and $\overline{\text{AUDRST}}$

Invalid only when being set to be pulled up/down via the I/O port (refer to section 24, I/O Ports):

PB1, PB3, PB5, PB8, PC9, PF15, PJ1, PJ3, PJ6, PJ9, PK2, PK5, PK8, and PK11

21.5 Usage Notes

1. Once a command is set, it is not updated until a different command is issued. When using the same command repeatedly, temporarily set a command that has no effect on the LSI's operation (e.g., BYPASS command) and set the desired command again.
2. No commands are accepted in hardware standby mode.
3. Whether or not the JTAG interface is in use, be sure to hold the $\overline{\text{TRST}}$ signal at the low level when resetting the chip. Specifically, hold the $\overline{\text{TRST}}$ signal low for 20 TCK clock cycles. For details, see section 34, Electrical Characteristics.
4. Up to 2 MHz frequency can be input to the TCK. Normal operation cannot be guaranteed if the frequency exceeds 2 MHz. For details, see section 34, Electrical Characteristics.
5. If data is transferred serially to the register that connects between the TDI and TDO pins and the data has a larger number of bits than the number of bits in that register, serial data bits that are in excess of the number of bits in the register are output from the TDO pin. In this case, such excess bits are from the data that has been input via the TDI pin.
6. If the serial transfer sequence failed, be sure to reset the JTAG interface using the $\overline{\text{TRST}}$ pin. Here, repeat the data transfer in question from the beginning regardless of the point at which the transfer failed.
7. The TDO pin starts outputting data at the falling edge of the TCK signal.
8. For easy debugging, design the $\overline{\text{TRST}}$ signal line pattern on the board so that it can be easily cut.
9. Do not halt this module by using the module standby function during boundary-scan. Instead, put the module into sleep mode immediately after a reset, then perform boundary scan. Refer to section 31, Power-Down Modes for details on the module standby function and the sleep mode.

Section 22 Advanced User Debugger II (AUD-II)

AUD-II provides functions for debugging the user program with this LSI mounted on the board. Using AUD-II facilitates configuration of a simple emulation system having functions such as acquisition of AUD tracing data and monitoring or tuning the contents of the on-chip RAM.

22.1 Features

AUD-II has the following two modes which can be switched by the setting of the AUDMD pin.

- AUD tracing mode
- RAM monitoring mode

(1) AUD Tracing Mode

- Eight input/output pins
- Branch tracing function

  Tracing both branch source and branch destination or tracing either branch source or branch destination can be selected
- Window data tracing function

  Window A and window B are available. When memory in a window area is accessed by the CPU or DMAC, the address and data are traced. Data lengths of 8, 16, and 32 bits are supported. M bus and I bus can be traced.
- Only the lower bits of an address, which are the difference from the previously accessed address, are output for efficient use of the AUDATA pins.
- Eight-stage FIFO
- Full tracing function

  All traced data is output even when the amount of information to be output exceeds the amount of traced information as making the CPU stall.
- Realtime tracing function

  The amount of traced information is reduced so that the CPU does not stall.
- AUD output clock selection

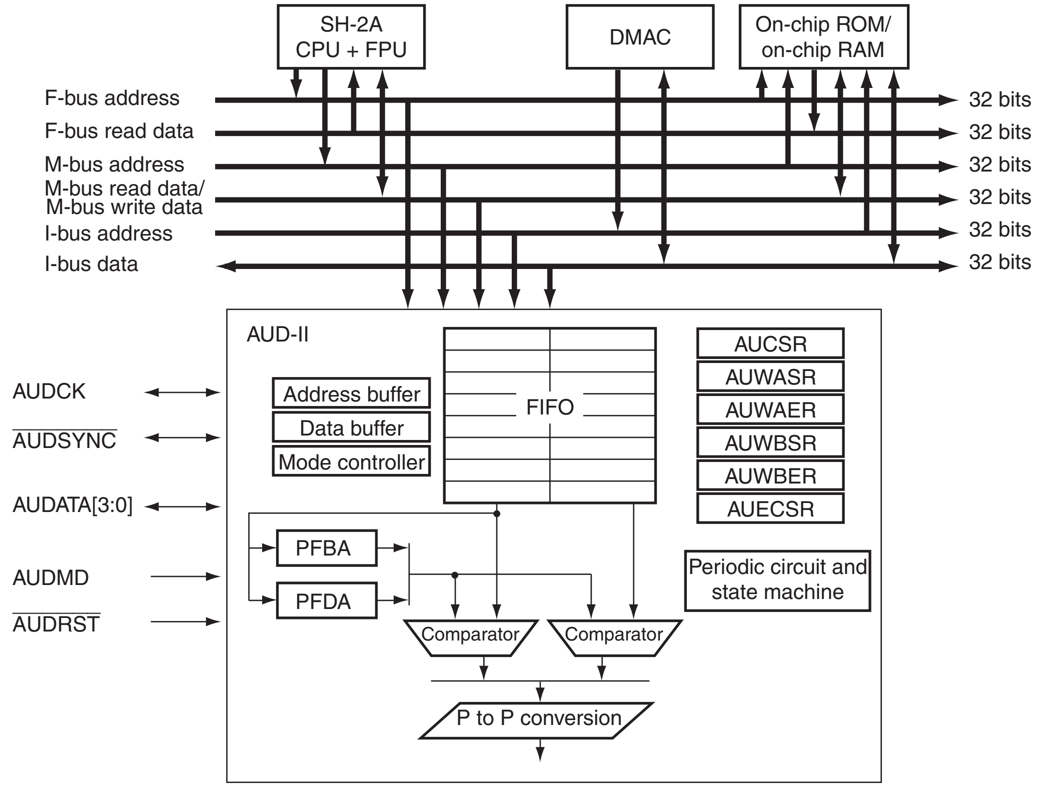
  The frequency of the AUD output clock is 1/4, 1/8, or 1/10 of that of the internal clock (ϕ).

(2) RAM Monitor Mode

- All memory-mapped modules connected to internal or external buses can be read from or written to.
- When an address is input on the AUDATA pins, data at the address is output.
- When an address and data are input on the AUDATA pins, the data is written to the address.

Note: In this section, AUDATA[3:0] denotes AUDATA 3 to AUDATA 0.

Figure 22.1 is a block diagram of AUD-II.



[Legend]
AUCSR: AUD Control Register
AUWASR: AUD Window A Start Address Register
AUWAER: AUD Window A End Address Register
AUWBSR: AUD Window B Start Address Register
AUWBER: AUD Window B End Address Register
AUECSR: AUD Extended Control Register

Figure 22.1 Block Diagram of AUD-II

22.2 Input/Output Pins

Table 22.1 shows the pin configuration of AUD-II.

Table 22.1 Pin Configuration

		Description	
Pin Name	**Symbol**	**AUD Tracing Mode**	**RAM Monitoring Mode**
AUD reset	$\overline{\text{AUDRST}}$	Input pin for an AUD-II reset signal	Input pin for an AUD-II reset signal
AUD sync	$\overline{\text{AUDSYNC}}$	Output pin for the data start position identification signal	Input pin for the data start position identification signal
AUD clock	AUDCK	Output pin for the clock signal	Input pin for the external clock signal
AUD mode	AUDMD	Input pin for the mode selecting signal (low)	Input pin for the mode selecting signal (high)
AUD data	AUDATA[3:0]	Output pins for traced data and addresses	Input pins for monitoring addresses or input/output pins for data

22.2.1 Commonly Used Pins

Table 22.2 Commonly Used Pins

Symbol	Description
AUDMD	The following mode can be selected by the level on this pin. Low: AUD tracing mode High: RAM monitoring mode The level on this pin must be changed while the $\overline{\text{AUDRST}}$ signal is low. When this pin is not connected, it is pulled up internally.
$\overline{\text{AUDRST}}$	While a low level is input on this pin, AUD-II is in the reset state. Buffers and internal states of AUD-II are initialized. When the level on the AUDMD pin is driven high after it has been stable, AUD-II starts in the selected mode. When this pin is not connected, it is pulled down internally.

22.2.2 Pin Descriptions in AUD Tracing Mode

Table 22.3 Pin Descriptions in AUD Tracing Mode

Symbol	Description
AUDCK	Outputs the clock signal to synchronize the AUDATA signals. The frequency of this clock signal is 1/4, 1/8, or 1/10 of the frequency of the internal clock (ϕ).
$\overline{\text{AUDSYNC}}$	Indicates the AUD bus command being output. 1: Indicates that AUD bus command 1 (CMD1) is being output 0: Indicates that AUD bus command 2 (CMD2), addresses, and data are being output Regardless of the data, a level of 0 is always output in the next cycle in which the LOST command is output.
AUDATA[3:0]	These pins output the following information at a specified timing. • AUD bus command • Branch source and destination addresses • Address and data for window tracing

22.2.3 Pin Descriptions in RAM Monitoring Mode

Table 22.4 Pin Descriptions in RAM Monitoring Mode

Symbol	Description
AUDCK	Input the external clock signal to be used for debugging. The input frequency must be less than the frequency from the EXTAL pin. When this pin is not connected, it is pulled up internally.
$\overline{\text{AUDSYNC}}$	Input the signal which indicates that the AUD bus command is valid. 1: While this signal is driven high, read data is output 0: While this signal is driven low, a write address, write data, and a DIR command can be input and the ready code is output When this pin is not connected, it is pulled up internally. Note: Assert this signal after a required command has been ready on the AUDATA pins. For details, see section 22.4, RAM Monitoring Mode.
AUDATA[3:0]	These pins output the following information at a specified timing. • AUD bus command • Addresses • Data Input a command on these pins. Then, the information is output after the ready code is output. When the $\overline{\text{AUDSYNC}}$ signal is negated, the information starts to be output. For details, see section 22.4, RAM Monitoring Mode. When this pin is not connected, it is pulled up internally.

22.3 AUD Tracing Mode

22.3.1 Register Descriptions

For controlling the various functions of AUD-II, AUD-II has registers AUCSR and AUECSR. For setting windows, registers AUWASR, AUWBSR, AUWAER, and AUWBER are available. Table 22.5 shows details of the register configuration.

Table 22.5 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
AUD control register	AUCSR	R/W	H'0000	H'FFFC0C00	8, 16
AUD window A start address register	AUWASR	R/W	Undefined	H'FFFC0C04	8, 16, 32
AUD window A end address register	AUWAER	R/W	Undefined	H'FFFC0C08	8, 16, 32
AUD window B start address register	AUWBSR	R/W	Undefined	H'FFFC0C0C	8, 16, 32
AUD window B end address register	AUWBER	R/W	Undefined	H'FFFC0C10	8, 16, 32
AUD extended control register	AUECSR	R/W	H'0000	H'FFFC0C14	8, 16

22.3.2 AUD Control Register (AUCSR)

AUCSR is a 16-bit readable/writable register.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CLK[1:0]		BW[1:0]		OC[1:0]		BR[1:0]		WA[1:0]		WB[1:0]		-	TM	-	EN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	CLK[1:0]	00	R/W	AUD Clock Select These bits set the ratio of the frequency of the clock signal output on the AUDCK pin to the internal clock (ϕ). For restrictions on the AUDCK clock ratio, see a description of Rules on AUDCK, under section 22.3.9 (2), Rules on AUDCK. 00: A ratio of 1 to 8 01: A ratio of 1 to 4 10: A ratio of 1 to 10 11: Reserved Note: CLK[1:0] must be changed only when the EN bit in AUCSR is 0.
13, 12	BW[1:0]	00	R/W	AUD Output Bus Width These bits specify the bus width for the AUDATA pins. These bits are provided for future use. So they are not yet actually implemented. Similar to other reserved bits, when writing to either of these bits, only 0 should be written to. Both of these bits are read as 0. 00: 4-bit mode. Trace information is output on pins AUDATA[3:0]. 01: Reserved 10: Reserved 11: Reserved
11, 10	OC[1:0]	00	R/W	Output Counter Mode Traced information to be output, such as information for branch tracing, includes only the lower address bits. However, all the 32 bits of the address can be output at specified intervals. The interval can be set in these bits. When these bits are set to B'11, only the difference from the previous traced address is output on the AUDATA pins other than the first timer after a reset. When these bits are cleared to B'00, all the address bits are output once per 128 times. 00: All the address bits are output once per 128 times 01: Reserved 10: Reserved 11: The lower bits changed from the previous address

Bit	Bit Name	Initial Value	R/W	Description
9, 8	BR[1:0]	00	R/W	Branch Tracing function AUD-II traces the branch destination and/or source addresses according to the setting of these bits. 00: Branch tracing disabled 01: Branch tracing enabled; branch source and destination are output 10: Branch tracing enabled; only branch source is output 11: Branch tracing enabled; only branch destination is output The setting in these bits takes effect when data is output after data next to the current data has been output.
7, 6	WA[1:0]	00	R/W	Window A Data Tracing Function AUD-II traces memory access to the area specified as window A according to the setting of these bits. As tracing conditions, either read access or write access, or both can be specified. Because multiple buses are in this LSI, the bus to be traced can be selected. For details, see section 22.3.7, AUD Extended Control Register (AUECSR). 00: Window A data tracing function disabled 01: Only write access is to be traced 10: Only read access is to be traced 11: Read and write accesses are to be traced
5, 4	WB[1:0]	00	R/W	Window B Data Tracing Function AUD-II traces memory access to the area specified as window B according to the setting of these bits. As tracing conditions, either read access or write access, or both can be specified. Because multiple buses are in this LSI, the bus to be traced can be selected. For details, see section 22.3.7, AUD Extended Control Register (AUECSR). 00: Window B data tracing function disabled 01: Only write access is to be traced 10: Only read access is to be traced 11: Read and write accesses are to be traced

Bit	Bit Name	Initial Value	R/W	Description
3	—	0	R/W	Reserved This bit is always read as 0. The write value should always be 0.
2	TM	0	R/W	Tracing mode Specifies the operation of the CPU when the FIFO buffer for storing various trace information on AUD-II is full. 0: Full tracing mode in which all traced events are output 1: Realtime tracing mode in which realtime tracing information is output without stopping the CPU
1	—	0	R/W	Reserved This bit is always read as 0. The write value should always be 0.
0	EN	0	R/W	AUD Trace Enable Enables tracing function specified by bits BR[1:0], WA[1:0], and WB[1:0]. When this bit is 0, the tracing function specified by bits BR[1:0], WA[1:0], and WB[1:0] bits is not executed. 0: AUD tracing function disabled 1: AUD tracing function enabled

22.3.3 AUD Window A Start Address Register (AUWASR)

AUWASR is a 32-bit readable/writable register that sets the logical address of the start address of window A to be traced. The end address of window A is specified by the AUD window A end address register (AUWAER). Window A is an area from the address specified by AUWASR to the address specified by AUWAER. AUWASR ≤ window A ≤ AUWAER.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

22.3.4 AUD Window A End Address Register (AUWAER)

AUWAER is a 32-bit readable/writable register that sets the logical address of the end address of window A to be traced. The start address of window A is specified by the AUD window A start address register (AUWASR).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

22.3.5 AUD Window B Start Address Register (AUWBSR)

AUWBSR is a 32-bit readable/writable register that sets the logical address of the start address of window B to be traced. The end address is specified by the AUD window B end address register (AUWBER). Window B is an area from the address specified by AUWBSR to the address specified by AUWBER. AUWBSR ≤ window B ≤ AUWBER.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

22.3.6 AUD Window B End Address Register (AUWBER)

AUWBER is a 32-bit readable/writable register that sets the logical address of the end address of window B to be traced. The start address of window B is specified by the AUD window B start address register (AUWBSR).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

22.3.7 AUD Extended Control Register (AUECSR)

AUECSR is a 16-bit readable/writable register.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	WA0B[2:0]			WB0B[2:0]			TREX	TRSB	TRGN	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 10	—	All 0	R/W	Reserved These bits are always read as 0. The write value should always be 0.
9 to 7	WA0B[2:0]	000	R/W	Window A Trace Bus Select These bits specify the internal bus to be traced. 000: Reserved 001: M bus 010: Reserved 011: Reserved 100: I bus 101: Reserved 110: Reserved 111: Reserved Note that the WA0B[2:0] bits must be modified when the EN bit in AUCSR is 0.

Bit	Bit Name	Initial Value	R/W	Description
6 to 4	WB0B[2:0]	000	R/W	Window B Trace Bus Select These bits specify the internal bus to be traced. 000: Reserved 001: M bus 010: Reserved 011: Reserved 100: I bus 101: Reserved 110: Reserved 111: Reserved Note that the WB0B[2:0] bits must be modified when the EN bit in AUCSR is 0.
3	TREX	0	R/W	Exception Branch Trace Select When the branch tracing function is enabled, this bit specifies whether or not an exception branch is to be traced. 0: Exception branches traced 1: Exception branches not traced
2	TRSB	0	R/W	Subroutine Branch Trace Select When the branch tracing function is enabled, this bit specifies whether or not a subroutine branch is to be traced. 0: Subroutine branches traced 1: Subroutine branches not traced

Bit	Bit Name	Initial Value	R/W	Description
1	TRGN	0	R/W	General Branch Trace Select When the branch tracing function is enabled, this bit specifies whether or not a general branch is to be traced. 0: General branches traced 1: General branches not traced Note: Branches are broken down into categories as follows: Exception branch: general illegal instruction, slot illegal instruction, bank underflow, interrupt operation, TRAPA instruction, RTE instruction, UBC break (PC), and power-on reset Sub-routine branch: BSR, BSRF, JSR, JSR/N, RTS, RTS/N, and RTV/N instructions General branch: BF, BT, BF/S, BT/S, BRA, BRAF, and JMP instructions
0	—	0	R/W	Reserved This bit is always read as 0. The write value should always be 0.

22.3.8 Operation

To use the AUD tracing function, perform the following procedure in this order.

1. While the $\overline{\text{AUDRST}}$ signal is asserted, negate the AUDMD signal.
2. Negate the $\overline{\text{AUDRST}}$ signal.
3. Set AUCSR, AUWASR, AUWAER, AUWBSR, AUWBER, and AUECSR.
4. Set the EN bit in AUCSR to 1.

AUD-II supports a branch tracing and a window data tracing functions. These functions can be executed independently. Further, for external output of the data traced by these functions, the user can select the realtime tracing mode or full tracing mode. Detailed operations of these AUD tracing modes are described below.

(1) AUD Bus Command

The information traced by AUD-II is output in a packet format on pins AUDATA[3:0] and $\overline{\text{AUDSYNC}}$ in synchronization with the AUDCK signal. The packet is comprised of a command part and a data part that contains 0 to 16 units of data. Packets are output continuously on pins AUDATA[3:0]. Normally, the command part is comprised of CMD1, which indicates the type of packet, and CMD2, which indicates the packet length of the data part. In addition, there is a special packet comprised solely of CMD1 that indicates the status of AUD-II. The $\overline{\text{AUDSYNC}}$ signal goes high when the bus is idle or when CMD1 is output on pins AUDATA[3:0]; and it goes low when CMD2, addresses, and data are output. Table 22.6 shows details of commands.

Table 22.6 List of AUD Bus Commands

Command	CMD1	CMD2	Descriptions
STDBY	B'0000	—	This command indicates the standby status. There is no subsequent data part to be output.
LOST	B'0001	—	In realtime tracing mode, this command indicates that the data to be output has been lost. In full tracing mode, this command indicates that the FIFO for AUD-II is full or the CPU stops temporarily because multiple tracing occurs simultaneously.
BGC	B'0010	(sda)(ssa)	Indicates that branch tracing information (general branching) is to be output. In the data part following this command, a branch destination address and a branch source address are output in the order on pins AUDATA[3:0]. sda: Indicates the size of a branch destination address. Disused when the tracing is only for the branch source. ssa: Indicates the size of a branch source address. Disused when the tracing is only for the branch destination. sda/ssa = B'00: The lower 4 bits of the address sda/ssa = B'01: The lower 8 bits of the address sda/ssa = B'10: The lower 16 bits of the address sda/ssa = B'11: All the 32-bits of address (full address)
BSC	B'0011	(sda)(ssa)	Indicates that branch tracing information (subroutine branching) is to be output. In the data part following this command, a branch destination address and a branch source address are output in the order on pins AUDATA[3:0]. sda: Indicates the size of a branch destination address. ssa: Indicates the size of a branch source address. sda/ssa = B'00: The lower 4 bits of the address sda/ssa = B'01: The lower 8 bits of the address sda/ssa = B'10: The lower 16 bits of the address sda/ssa = B'11: All the 32-bits of address (full address)

Command	CMD1	CMD2	Descriptions
BEC	B'0100	(sda)(ssa)	Indicates that branch tracing information (exception branching) is to be output. In the data part following this command, a branch destination address and a branch source address are output in the order on pins AUDATA[3:0]. sda: Indicates the size of a branch destination address. ssa: Indicates the size of a branch source address. sda/ssa = B'00: The lower 4 bits of the address sda/ssa = B'01: The lower 8 bits of the address sda/ssa = B'10: The lower 16 bits of the address sda/ssa = B'11: All the 32-bits of address (full address)
WDWM	B'1000	(sa)(sd)	Indicates that stored information of M-bus data tracing in accessing the window area. In the data part following this command, a stored address and stored data are output in the order on pins AUDATA[3:0]. sa: Indicates the size of a stored address. B'00: The lower 4 bits of the address B'01: The lower 8 bits of the address B'10: The lower 16 bits of the address B'11: All the 32 bits of address (full address) sd: Indicates the size of stored data. B'01: Byte data (8 bits) B'10: Word data (16 bits) B'11: Longword data (32 bits)
WDWI	B'1001	(sa)(sd)	Indicates that stored information of I-bus data tracing in accessing the window area. In the data part following this command, a stored address and stored data are output in the order on pins AUDATA[3:0]. sa: Indicates the size of a stored address. B'00: The lower 4 bits of the address B'01: The lower 8 bits of the address B'10: The lower 16 bits of the address B'11: All the 32 bits of address (full address) sd: Indicates the size of stored data. B'01: Byte data (8 bits) B'10: Word data (16 bits) B'11: Longword data (32 bits)

Command	CMD1	CMD2	Descriptions
WDRM	B'1101	(sa)(sd)	Indicates that information of M-bus data tracing in reading the window area. In the data part following this command, a read address and read data are output in the order on pins AUDATA[3:0]. sa: Indicates the size of a read address. B'00: The lower 4 bits of the address B'01: The lower 8 bits of the address B'10: The lower 16 bits of the address B'11: All the 32 bits of address (full address) sd: Indicates the size of read data. B'01: Byte data (8 bits) B'10: Word data (16 bits) B'11: Longword data (32 bits)
WDRI	B'1110	(sa)(sd)	Indicates that information of I-bus data tracing in reading the window area. In the data part following this command, a read address and read data are output in the order on pins AUDATA[3:0]. sa: Indicates the size of a read address. B'00: The lower 4 bits of the address B'01: The lower 8 bits of the address B'10: The lower 16 bits of the address B'11: All the 32 bits of address (full address) sd: Indicates the size of read data. B'01: Byte data (8 bits) B'10: Word data (16 bits) B'11: Longword data (32 bits)

(2) Branch Tracing

The branch tracing function traces events in which the PC changes due to the execution of a branch instruction or the occurrence of an interrupt, and outputs the addresses of a branch source and a branch destination to an external device. Set the BR[1:0] and EN bits in AUCSR and the TREX, TRSB, and TRGN bits in AUECSR to execute the branch tracing of a user program.

Figures 22.2 to 22.4 show examples of branch tracing output.

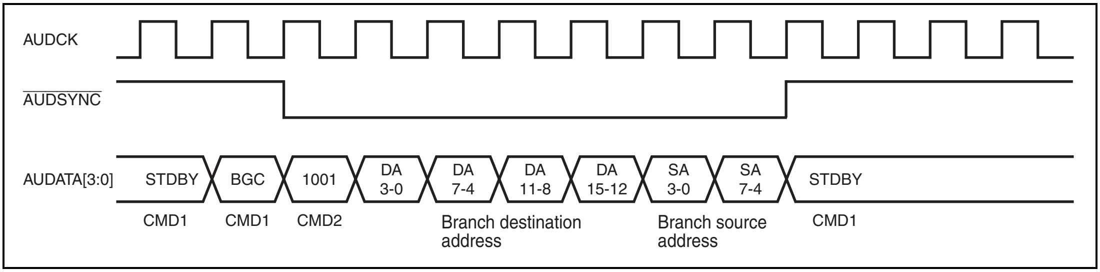


Figure 22.2 Branch Tracing using BGC Command (Branch Destination and Source) (Lower 16 Bits of Branch Destination Address; Lower 8 Bits of Branch Source Address)

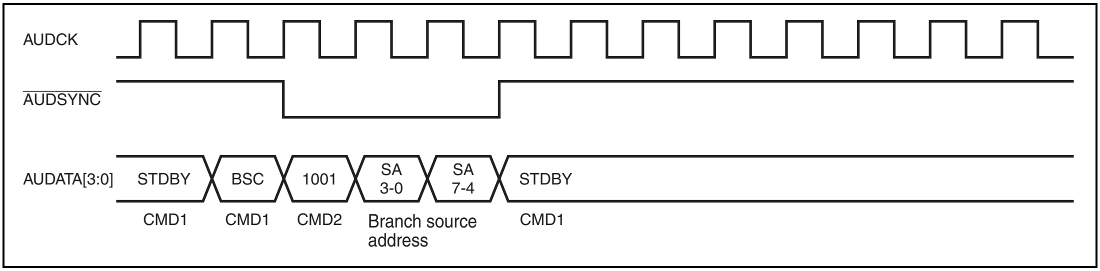


Figure 22.3 Branch Tracing using BSC Command (Branch Source Only) (Lower 8 Bits of Branch Source Address)

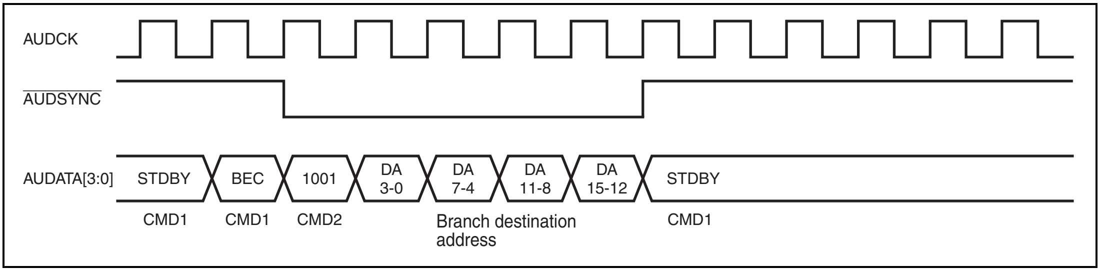


Figure 22.4 Branch Tracing using BEC Command (Branch Destination Only) (Lower 16 Bits of Branch Destination Address)

When both a branch destination and branch source are output, address differences are compared with the previous branch destination address. When only a branch source is output, address differences are compared with the previous branch source address. When only a branch destination is output, address differences are compared with the previous branch destination address.

In a user program, no branch information is output as long as a branching event (branch instruction execution or an interrupt) does not occur (the STDBY command is output instead). When a branch event occurs while the source and destination addresses are both set to be output, following the command, a branch destination address and a branch source address are output in the order on pins AUDATA[3:0]. The sizes of the addresses (4, 8, 16, or 32 bits) to be output are determined by comparing the addresses with the previously output branch destination address (PFBA).

This algorithm is shown in figure 22.5.

Definition:

PFBA: Previously accessed address of branch destination (all 32 bits stored)
(when only branch source is traced, the branch source address is stored)

CDA: Branch destination address to be output

CSA: Branch source address to be output

```
if(PFBA[31:0] == 0x00000000) {output32bit(CDA[31:0]);}
else if(PFBA[31:4] == CDA[31:4]) {output4bit(CDA[3:0]);}
else if(PFBA[31:8] == CDA[31:8]) {output8bit(CDA[7:0]);}
else if(PFBA[31:16] == CDA[31:16]) {output16bit(CDA[15:0]);}
else {output32bit(CDA[31:0]);}

if(PFBA[31:0] == 0x00000000) {output32bit(CSA[31:0]);}
else if(PFBA[31:4] == CSA[31:4]) {output4bit(CSA[3:0]);}
else if(PFBA[31:8] == CSA[31:8]) {output8bit(CSA[7:0]);}
else if(PFBA[31:16] == CSA[31:16]) {output16bit(CSA[15:0]);}
else {output32bit(CSA[31:0]);}

PFBA = CDA;  /* update PFBA */
```

Figure 22.5 Branch Tracing Algorithm

The use of this algorithm can substantially reduce the amount of traced data to be output. window data tracing information is accumulated as long as there is available space in the FIFO. The operation when the FIFO is full depends on the TM bit (tracing mode) in AUCSR.

- Branch Source Address

  The address to which a branch source address (TSA) points depends on the type of branches as follows:

  — Branch instructions (general and subroutine branches)

    The TSA points to the address of the branch instruction.

- Instruction-asynchronous exception (16-bit and 32-bit instructions)

  The TSA points to the address of the instruction that has been replaced by the exception handling.

- Instruction-synchronous exception (the TRAPA and RTE instructions)

  The TSA points to the address of the instruction.

Table 22.7 Definition of Branch Source Addresses

Type	General Branching	Subroutine Branching	Instruction Asynchronous Exception (16 Bits)	Instruction Asynchronous Exception (32 Bits)	Instruction Synchronous Exception
1000	BRA	BSR	NOP*[1]	32-bit instruction*[1]	TRAPA
1002	NOP	NOP	NOP*[2]		NOP
1004	NOP	NOP	NOP	NOP*[2]	NOP
1006	NOP	NOP	NOP	NOP	NOP
TSA	1000	1000	1002	1004	1000

Notes: 1. Indicates that the instruction has been executed.

2. Indicates that the instruction has been replaced by the exception handling.

(3) Window Data Tracing

The window data tracing function outputs traced information in accessing to a memory area (called window) specified by two address pointers. AUD-II can handle two window areas: window A specified by AUWASR and AUWAER and window B specified by AUWBSR and AUWBER. By setting the EN bit to 1 after setting the WA[1:0] and WB[1:0] bits in AUCSR, memory access information in a user program is traced.

Figure 22.6 shows an example of window data tracing.

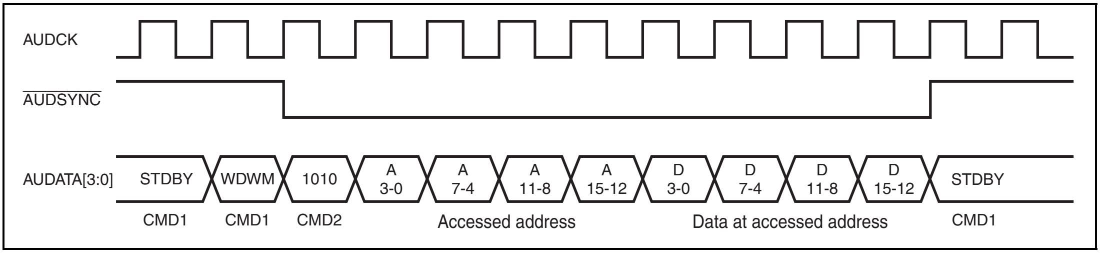


Figure 22.6 Window Data Tracing using WDWM Command (M Bus, Lower 16 Bits of Address, Word-Size Data)

When data in a window area is accessed in a user program, the command, accessed address, and the data at the accessed address are output in the order on pins AUDATA[3:0]. In a way similar to branch tracing, the address sizes (4, 8, 16, or 32 bits) to be output are determined by comparing the two addresses with the previously output address (PFDA). The data at the accessed address is output according to the access size (8, 16, or 32).

This algorithm is shown in figure 22.7.

Definition:

PFDA: Previously accessed address (all 32 bits stored)

CDA: Data to be output

```
/* PFDA : initial value = H'00000000 */
if(PFDA[31:0] == 0x00000000) {output32bit(CDA[31:0]);}
else if(PFDA[31:4] == CDA[31:4]) {output4bit(CDA[3:0]);}
else if(PFDA[31:8] == CDA[31:8]) {output8bit(CDA[7:0]);}
else if(PFDA[31:16] == CDA[31:16]) {output16bit(CDA[15:0]);}
else {output32bit(CDA[31:0]);}

PFDA = CDA;  /* update PFDA */
```

Figure 22.7 Window Data Tracing Algorithm

The use of this algorithm can substantially reduce the amount of traced data to be output. Branch tracing information is accumulated as long as there is available space in the FIFO. The operation when the FIFO is full depends on the TM bit (tracing mode) in AUCSR.

(4) Realtime Tracing Mode

Realtime tracing mode outputs various data traced by branch tracing or window data tracing in realtime to an external device via pins AUDATA[3:0]. Setting the TM bit in AUCSR to 1 selects this mode.

In this mode, the CPU operates in the same way as when the tracing function is not used, even during output of trace data. The traced events are stored as long as there is available space in the FIFO of AUD-II, and are output to an external device. When the FIFO is full, the events that should be traced are ignored, and they are not output externally. However, any missed event is notified to an external device by the LOST command. In the cycle following the LOST command output cycle, the $\overline{\text{AUDSYNC}}$ signal is driven low.

If multiple events occur simultaneously, corresponding information is stored in the FIFO in the following order: window data tracing (M bus > I bus) and branch tracing.

Figure 22.8 shows an example where multiple traced information is lost between two branch trace events that are output. Figure 22.9 shows an example of LOST and STDBY states following the LOST command.

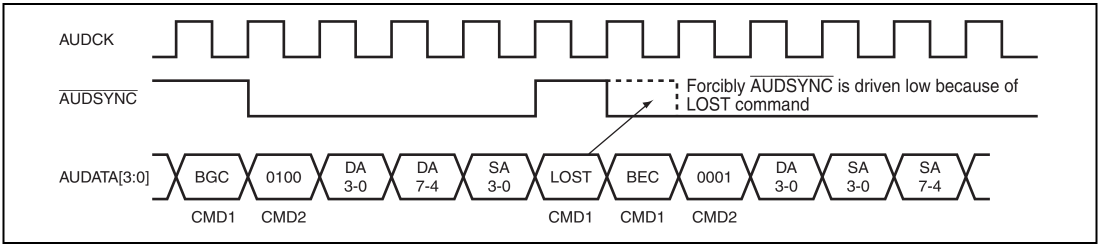


Figure 22.8 Example Where Trace Data is Lost during Realtime Tracing

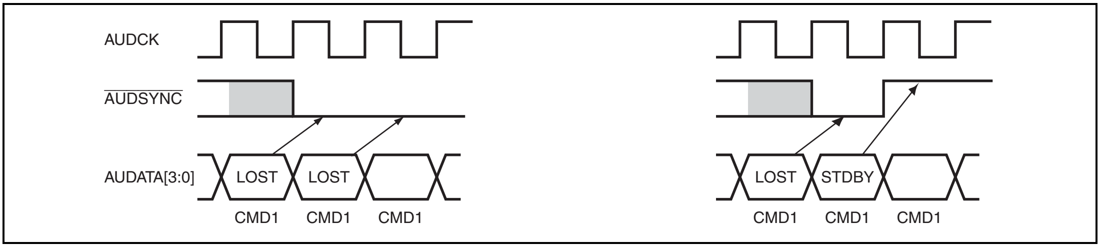


Figure 22.9 Example of LOST and STDBY States Following LOST Command

(5) Full Tracing Mode

In full tracing mode, all traced data is output externally, without losing it. This mode is selected by setting the TM bit in AUCSR to 0. Various types of traced information in full tracing mode are output externally through the FIFO in AUD-II. If the FIFO becomes full, the CPU stops operation until the traced data in the FIFO is output to an external device.

In this a case, unlike the realtime tracing mode, traced data is not lost. However, if the CPU temporarily stops operation because the FIFO is full, the LOST command indicating the temporary halting of the CPU is output. In a way similar to the realtime tracing mode, in the cycle following the LOST command output cycle, the $\overline{\text{AUDSYNC}}$ signal is driven low as shown in figure 22.9.

(6) Address Comparison and Output of Address Differences

For the efficient use of the narrow-bit-width AUDATA output, only the necessary lower bits of traced information are output in the address part. The lower bits are obtained by comparing the traced address with the previously output address stored in PFBA and PFDA.

The sizes (4, 8, 16, or 32 bits) of the branch destination and source addresses are determined by comparing the two addresses with PFBA. Similarly, the address to be output by window data tracing is determined by comparing the two addresses with PFDA. PFBA and PFDA used to compare addresses are updated each time the compared traced information is output.

PFBA and PFDA are initialized to H'00000000 (called the disabled state) when the EN bit in AUCSR is changed from 0 to 1, or when the output counter overflows. When PFBA and PFDA are disabled, 32 bits of the address are output in the address part of traced information. The output counter is incremented each time a trace event occurs, and disables PFBA and PFDA once per 128 times a trace event occurs. In addition, the output counter is cleared to 0 when the EN bit in AUCSR is changed from 0 to 1. The output counter functions can be used for recovery of the corrupted traced information.

Figure 22.10 shows an example of address comparison.

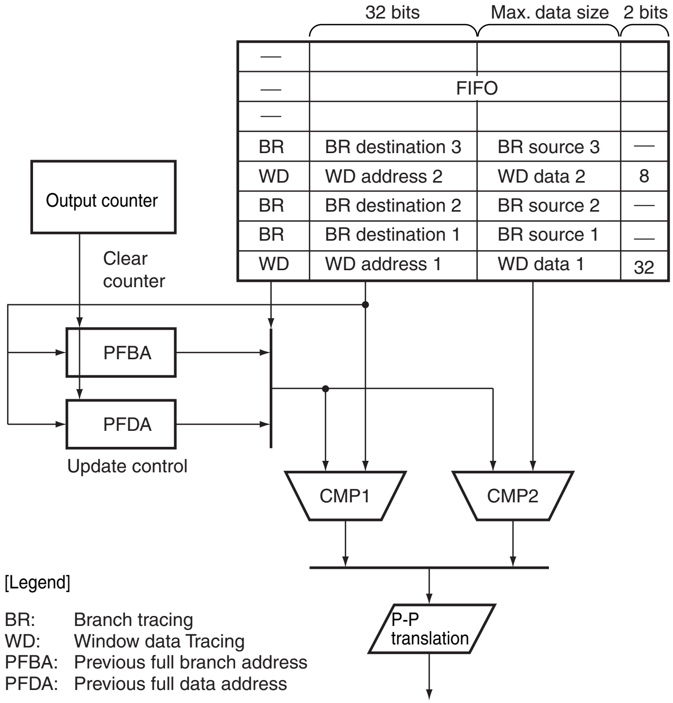

32 bits
Max. data size
2 bits
—
— FIFO
—
BR | BR destination 3 | BR source 3 | —
WD | WD address 2 | WD data 2 | 8
BR | BR destination 2 | BR source 2 | —
BR | BR destination 1 | BR source 1 | —
WD | WD address 1 | WD data 1 | 32
Output counter
Clear counter
PFBA
PFDA
Update control
CMP1
CMP2
P-P translation
[Legend]
BR: Branch tracing
WD: Window data Tracing
PFBA: Previous full branch address
PFDA: Previous full data address

Figure 22.10 Example of Address Comparison

1. First, WD address 1 and PFDA are matched and compared in CMP1. Based on the results of the comparison, the lower part (4, 8, 16, or 32 bits) of WD address 1 is output. Next, WD data 1, 32-bit long, is output in its native length. Subsequently, PFDA is updated to WD address 1.
2. In the next step, BR destination 1 and PFBA are compared in CMP1, and BR source 1 and PFBA are compared in CMP2. Based on the results of the comparison in CMP1, the lower part (4, 8, 16, or 32 bits) of BR destination 1 is output. Next, the lower part (4, 8, 16, or 32 bits) of BR source 1 is output based on the results of comparison in CMP2. Subsequently, PFBA is updated to BR destination 1.
3. BR destination 2 and PFBA are compared in CMP1, and BR source 2 and PFBA are compared in CMP2. PFBA holds BR destination 1, which was stored in step 2 above. According to the results of the comparison in CMP1, the lower part (4, 8, 16, or 32 bits) of BR destination 2 is output. Next, the lower part (4, 8, 16, or 32 bits) of BR source 2 is output according to the results of comparison in CMP2. Subsequently, PFBA is updated to BR destination 2.

4. WD address 2 and PFDA are compared in CMP1. PFDA holds WD address 1, which was stored in step 1 above. Based on the results of the comparison, the lower part (4, 8, 16, or 32 bits) of WD address 2 is output. Next, WD data 2, which is 8 bits long, is output in its native bit length. Subsequently, PFDA is updated to WD address 2.
5. BR destination 3 and PFBA are compared in CMP1, and BR source 3 and PFBA are compared in CMP2. PFBA holds BR destination 2, which was stored in step 3 above. Based on the results of the comparison in CMP1, the lower part (4, 8, 16, or 32 bits) of BR destination 3 is output. Next, based on the results of the comparison in CMP2, the lower part (4, 8, 16, or 32 bits) of BR source 3 is output. Subsequently, PFBA is updated to BR destination 3.
6. If the output counter overflows, the values in PFBA and PFDA are nullified. The branch destination, the branch source, and the data address part in the trace information are always output in a 32-bit length.

22.3.9 Usage Notes on AUD Tracing Mode

(1) Rules on Initialization of AUD Tracing Mode

AUD-II is initialized according to the following conditions.

The AUCSR and AUECSR registers are initialized to H'0000 by a power-on reset, a low level input on the $\overline{\text{AUDRST}}$ pin, and a module standby. At the same time, AUDATA[3:0], $\overline{\text{AUDSYNC}}$, and AUDCK output H'0, B'1, and B'1, respectively.

- Power-on reset
- A low level input on the $\overline{\text{AUDRST}}$ pin
- Module standby

  Note that the module standby mode is not entered under the following conditions.

  - When the EN bit in AUCSR is 1
  - When the traced data is in the FIFO
- The EN bit in AUCSR changes from 0 to 1.

  The FIFO of AUD-II is cleared (emptied), and PFBA, PFDA, and the output counter are initialized.
- The output counter overflows.

  The values in PFBA and PFDA are initialized (H'00000000).

(2) Rules on AUDCK

- When the EN bit in AUCSR is 1, the clock signal is output on the AUDCK pin. The clock signal is not output when the EN bit is 0. However, if the FIFO contains valid data, this state is attained after all data has been transmitted.
- The CK[1:0] bits in AUCSR must be changed when the EN bit in AUCSR is 0.
- The frequency of the AUDCK clock must not exceed the input frequency from the EXTAL pin. Settable AUDCK clock ratios are listed below:

Table 22.8 Settable AUDCK Clock Ratios

PLL Multiplication Ratio of Internal Clock (ϕ)	Allowable AUDCK Clock Ratio
×4	1/4, 1/8, 1/10
×6, ×8	1/8, 1/10
×10	1/10

(3) Writing to AUD-II Register

- The AUD-II registers are written to through the I bus. Therefore, in a cycle immediately after an AUD-II register is rewritten to change tracing conditions, tracing may not be performed according to the modified conditions. To ensure that the conditions in the AUD-II register are changed, execute more than five NOP instructions after reading the rewritten register once. After that tracing conditions will be reflected in the register.
- The AUD-II register must be changed when the EN bit in AUCSR is 0.

(4) Other Notes

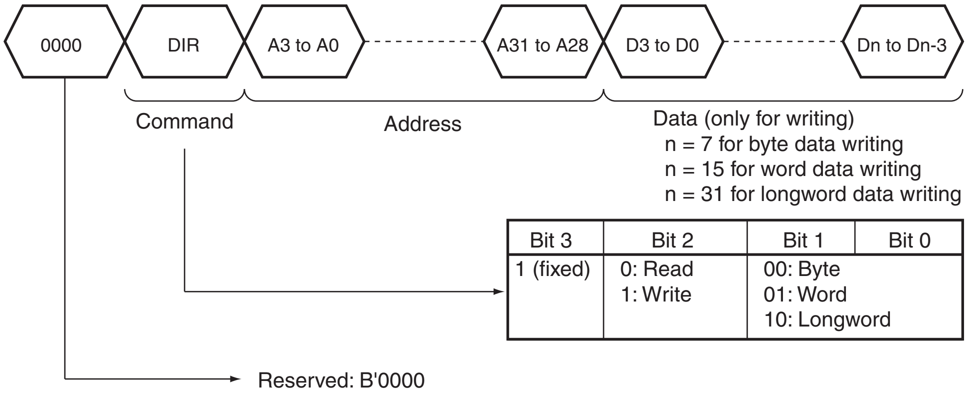
- If the FIFO of AUD-II contains valid data and the tracing function is disabled, the FIFO assumes the disabled state after the remaining data is output. The tracing function should not be enabled when the data remaining in the FIFO is being output. The tracing function of AUD-II should be enabled only after confirming that no data is being output on AUD-II.
- The AUD tracing function can be used in sleep mode.

22.4 RAM Monitoring Mode

In RAM monitoring mode, all memory-mapped modules connected to internal or external buses can be read from or written to. The contents of the on-chip RAM can be monitored or tuned.

22.4.1 Communication Protocol

AUD-II reads data on the AUDATA pins at an assertion of the $\overline{\text{AUDSYNC}}$ signal. Input the AUDATA signals in the following format.



Bit 3	Bit 2	Bit 1	Bit 0
1 (fixed)	0: Read 1: Write	00: Byte 01: Word 10: Longword	

Figure 22.11 AUDATA Input Format

22.4.2 Operations

To use the RAM monitoring function, negate the AUDMD signal while the $\overline{\text{AUDRST}}$ signal is asserted. Then negate the $\overline{\text{AUDRST}}$ signal. After that, AUD-II enters the RAM monitoring mode. Figure 22.12 shows an example of read operation and figure 22.13 shows an example of write operation.

Assert the $\overline{\text{AUDSYNC}}$ signal before data is input. Input a command, address, and data (required only for writing) in the format shown in figure 22.11. AUD-II starts reading from or writing to the specified address. AUD-II outputs the not-ready code (B'0000) during the internal processing and the ready code (B'0001) on completion of the internal processing (see figures 22.12 and 22.13). Table 22.9 shows the meaning of the code.

For reading, negate the $\overline{\text{AUDSYNC}}$ signal when detecting the ready code. AUD-II outputs the specified size of data (see figure 22.12). When an undefined command (i.e., other than a DIR command) is input, AUD-II handles the command as an error. In this case, AUD-II does not read or write and sets bit 1 in the code shown in table 22.9 to 1.

If a command specified by the DIR command causes a bus error, AUD-II does not execute read or write specified by the command and sets bit 2 in the code shown in table 22.9 to 1 (see figure 22.14).

The bus errors are shown below.

- When an address of 4n +1 or 4n + 3 is accessed in words
- When an address of 4n +1, 4n +2, or 4n + 3 is accessed in longwords
- When an address in the external space is accessed in single-chip mode

Table 22.9 Meaning of Each Bit of Code

Bit	Description
3	0 (fixed)
2	0: Normal state 1: A bus error has occurred
1	0: Normal state 1: A command error has occurred
0	0: Not ready 1: Ready

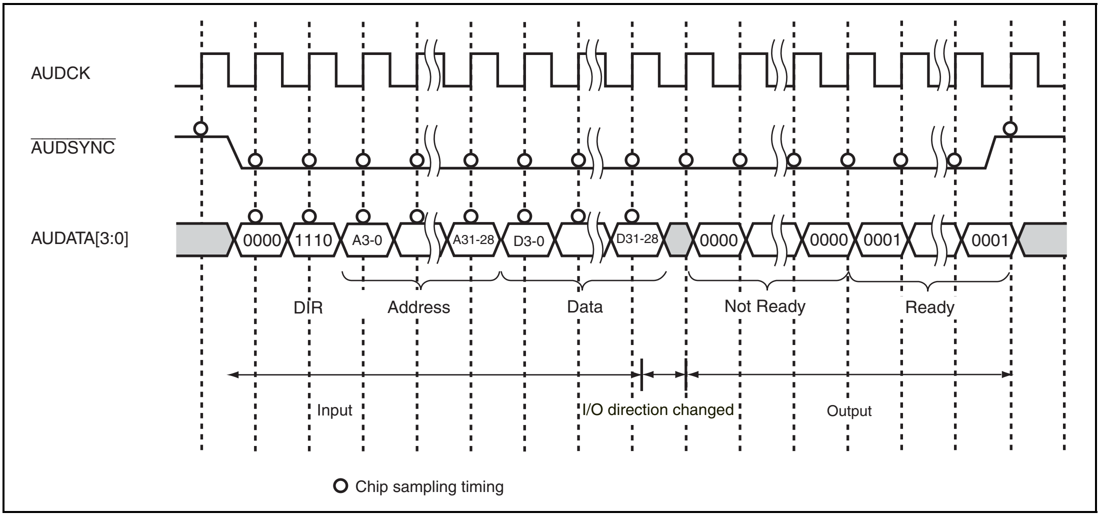


Figure 22.12 Example Operation of Byte Read

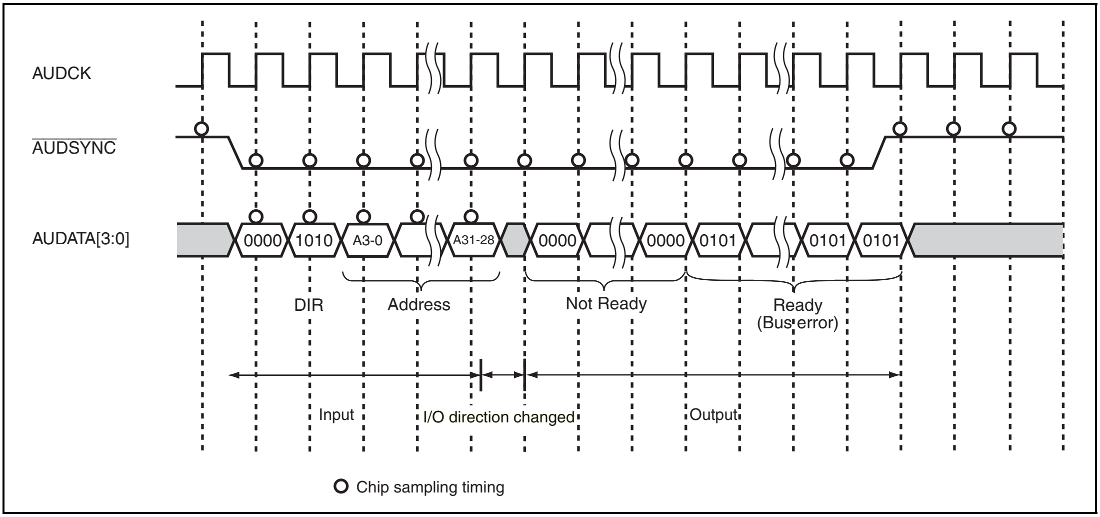


Figure 22.13 Example Operation of Longword Write

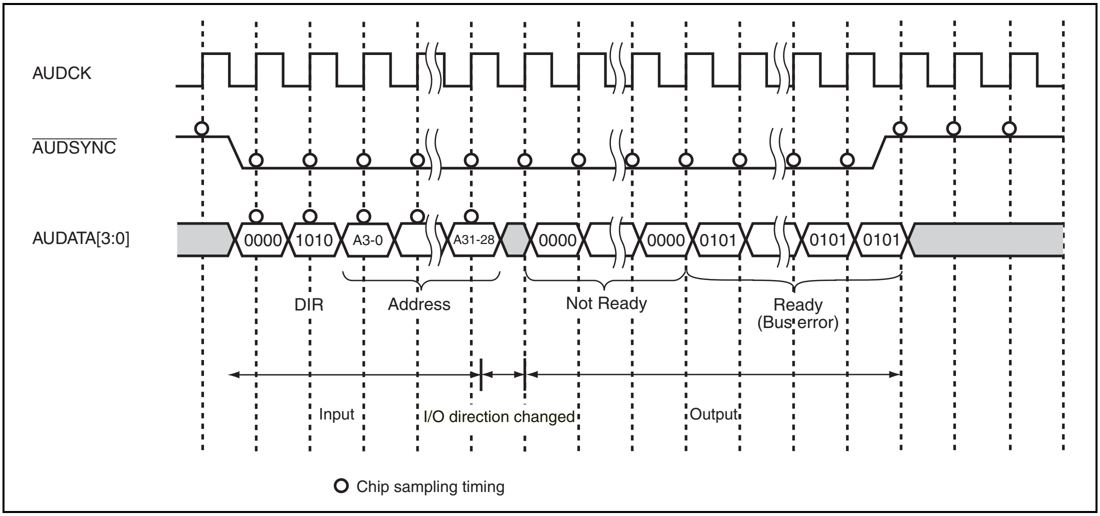


Figure 22.14 Example Operation of Error (Longword Read)

22.4.3 Notes on RAM Monitoring Mode

(1) Rules on Initialization in RAM Monitoring

Buffers and internal states of AUD-II are initialized under the following conditions.

- When a power-on reset is input
- When a low level is input on the $\overline{\text{AUDRST}}$ pin
- When a transition to the module standby mode is made

(2) Rules on AUDCK

AUDCK is for an external clock input in this mode. The frequency of the external clock input on the AUDCK pin must be less than that of the input clock from the EXTAL pin.

(3) Other Notes

Hold the $\overline{\text{AUDSYNC}}$ signal low until the ready code is returned after inputting a command on the AUDATA pins.

The RAM monitoring function can be used in sleep mode.

Section 23 Pin Function Controller (PFC)

The pin function controller (PFC) consists of the registers that control the functions and direction (I/O) of the multiplex pins. The multiplex pins of this LSI are summarized in tables 23.1 to 23.11.

Table 23.1 Multiplex Pin List (Port A)

Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)	Function 10 (Related Module)	Function 11 (Related Module)	Function 12 (Related Module)
PA0 (port)	A0 (BSC)	—	—	—	—	—
PA1 (port)	A1 (BSC)	—	—	—	—	—
PA2 (port)	A2 (BSC)	—	—	—	—	—
PA3 (port)	A3 (BSC)	—	—	—	—	—
PA4 (port)	A4 (BSC)	—	—	TOB00B (ATU-III)	—	—
PA5 (port)	A5 (BSC)	—	—	TOB01B (ATU-III)	—	—
PA6 (port)	A6 (BSC)	—	—	TOB02B (ATU-III)	TIF0A (ATU-III)	—
PA7 (port)	A7 (BSC)	—	—	TOB03B (ATU-III)	TIF1A (ATU-III)	—
PA8 (port)	A8 (BSC)	—	—	TOB10B (ATU-III)	TIF2A (ATU-III)	—
PA9 (port)	A9 (BSC)	—	—	TOB11B (ATU-III)	—	—
PA10 (port)	A10 (BSC)	—	—	TOB12B (ATU-III)	TIF0B (ATU-III)	—
PA11 (port)	A11 (BSC)	—	—	TOB13B (ATU-III)	TIF1B (ATU-III)	—
PA12 (port)	A12 (BSC)	—	—	TIA00 (ATU-III)	TIF2B (ATU-III)	—
PA13 (port)	A13 (BSC)	—	—	TIA01 (ATU-III)	—	—
PA14 (port)	A14 (BSC)	—	—	TIA02 (ATU-III)	—	—
PA15 (port)	A15 (BSC)	—	—	TIA03 (ATU-III)	—	—

Table 23.2 Multiplex Pin List (Port B)

Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)	Function 10 (Related Module)	Function 11 (Related Module)	Function 12 (Related Module)
PB0 (port)	A16 (BSC)	MOSIA (RSPI)	—	—	—	FRTxD_B (FlexRay)
PB1 (port)	A17 (BSC)	MISOA (RSPI)	—	—	—	FRRxD_B (FlexRay)
PB2 (port)	A18 (BSC)	MOSIB (RSPI)	—	TIA04 (ATU-III)	—	SSLA6 (RSPI)
PB3 (port)	A19 (BSC)	MISOB (RSPI)	—	TIA05 (ATU-III)	—	SSLA7 (RSPI)
PB4 (port)	A20 (BSC)	CTx_B (RCAN-TL1)	TIF6 (ATU-III)	TIF26 (ATU-III)	—	TxD_B (SCI)
PB5 (port)	A21 (BSC)	CRx_B (RCAN-TL1)	TIF7 (ATU-III)	TIF27 (ATU-III)	—	RxD_B (SCI)
PB6 (port)	$\overline{\text{WE0}}$ (BSC)	—	—	—	—	SCK_B (SCI)
PB7 (port)	$\overline{\text{WE1}}$ (BSC)	—	—	—	—	TxD_A (SCI)
PB8 (port)	$\overline{\text{WAIT}}$ (BSC)	TOE20 (ATU-III)	—	TIF20 (ATU-III)	—	RxD_A (SCI)
PB9 (port)	$\overline{\text{RD}}$ (BSC)	—	—	—	—	SSLB0 (RSPI)
PB10 (port)	$\overline{\text{CS0}}$ (BSC)	—	—	—	—	SSLB1 (RSPI)
PB11 (port)	$\overline{\text{CS1}}$ (BSC)	TOE21 (ATU-III)	—	TIF21 (ATU-III)	—	SSLB2 (RSPI)
PB12 (port)	$\overline{\text{CS2}}$ (BSC)	RSPCKA (RSPI)	—	—	—	FREN_B (FlexRay)
PB13 (port)	$\overline{\text{CS3}}$ (BSC)	RSPCKB (RSPI)	—	TIF22 (ATU-III)	—	—
PB14 (port)	RD/$\overline{\text{WR}}$ (BSC)	—	—	—	—	—

Table 23.3 Multiplex Pin List (Port C)

Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)	Function 10 (Related Module)	Function 11 (Related Module)	Function 12 (Related Module)
PC0 (port)	D0 (BSC)	—	—	—	—	—
PC1 (port)	D1 (BSC)	—	—	—	—	—
PC2 (port)	D2 (BSC)	—	—	—	—	—
PC3 (port)	D3 (BSC)	—	—	—	—	—
PC4 (port)	D4 (BSC)	—	—	—	—	—
PC5 (port)	D5 (BSC)	—	—	—	—	—
PC6 (port)	D6 (BSC)	—	—	—	—	ADTRG_A (ADC)
PC7 (port)	D7 (BSC)	—	—	—	—	ADEND_A (ADC)
PC8 (port)	D8 (BSC)	—	—	TOD00A (ATU-III)	TxD_A (SCI)	CTx_A (RCAN)
PC9 (port)	D9 (BSC)	—	—	TOD01A (ATU-III)	RxD_A (SCI)	CRx_A (RCAN)
PC10 (port)	D10 (BSC)	—	—	TOD02A (ATU-III)	SCK_A (SCI)	—
PC11 (port)	D11 (BSC)	—	—	TOD03A (ATU-III)	—	SSLB3 (RSPI)
PC12 (port)	D12 (BSC)	—	—	TOD10A (ATU-III)	—	SSLA0 (RSPI)
PC13 (port)	D13 (BSC)	—	—	TOD11A (ATU-III)	—	SSLA1 (RSPI)
PC14 (port)	D14 (BSC)	—	—	TOD12A (ATU-III)	—	SSLA2 (RSPI)
PC15 (port)	D15 (BSC)	—	—	TOD13A (ATU-III)	—	SSLA3 (RSPI)

Table 23.4 Multiplex Pin List (Port D)

Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)	Function 10 (Related Module)	Function 11 (Related Module)	Function 12 (Related Module)
PD0 (port)	TIOC00 (ATU-III)	TIOC31 (ATU-III)	—	—	—	—
PD1 (port)	TIOC01 (ATU-III)	TOE20 (ATU-III)	—	—	—	—
PD2 (port)	TIOC02 (ATU-III)	TOE21 (ATU-III)	TOE52 (ATU-III)	—	—	—
PD3 (port)	TIOC03 (ATU-III)	TOE22 (ATU-III)	TOE53 (ATU-III)	—	—	—
PD4 (port)	TIOC10 (ATU-III)	TIOC32 (ATU-III)	TOE52 (ATU-III)	—	—	—
PD5 (port)	TIOC11 (ATU-III)	TOE23 (ATU-III)	TOE40 (ATU-III)	—	—	—
PD6 (port)	TIOC12 (ATU-III)	—	TOE41 (ATU-III)	—	—	—
PD7 (port)	TIOC13 (ATU-III)	—	TOE42 (ATU-III)	—	—	—
PD8 (port)	TIOC20 (ATU-III)	TIOC33 (ATU-III)	TOE53 (ATU-III)	—	—	—
PD9 (port)	TIOC21 (ATU-III)	TIF0B (ATU-III)	TOE43 (ATU-III)	—	—	—
PD10 (port)	TIOC22 (ATU-III)	TIF1B (ATU-III)	TOE50 (ATU-III)	—	—	—
PD11 (port)	TIOC23 (ATU-III)	TIF2B (ATU-III)	TOE51 (ATU-III)	—	—	—
PD12 (port)	TCLKA (ATU-III)	TIOC41 (ATU-III)	TIJ0 (ATU-III)	—	—	—
PD13 (port)	TCLKB (ATU-III)	—	TIJ1 (ATU-III)	TOE60 (ATU-III)	—	—

Table 23.5 Multiplex Pin List (Port E)

Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)	Function 10 (Related Module)	Function 11 (Related Module)	Function 12 (Related Module)
PE0 (port)	TIA00 (ATU-III)	—	—	—	—	—
PE1 (port)	TIA01 (ATU-III)	TIOC42 (ATU-III)	TIOC40 (ATU-III)	—	—	—
PE2 (port)	TIA02 (ATU-III)	TIOC43 (ATU-III)	TIOC30 (ATU-III)	—	—	—
PE3 (port)	TIA03 (ATU-III)	—	—	TOE61 (ATU-III)	TIF23 (ATU-III)	—
PE4 (port)	TIA04 (ATU-III)	—	—	TOE62 (ATU-III)	TIF24 (ATU-III)	—
PE5 (port)	TIA05 (ATU-III)	—	—	TOE63 (ATU-III)	TIF25 (ATU-III)	—
PE6 (port)	TOE00 (ATU-III)	CTx_B (RCAN-TL1)	—	—	—	—
PE7 (port)	TOE01 (ATU-III)	CRx_B (RCAN-TL1)	—	—	—	—
PE8 (port)	TOE02 (ATU-III)	—	—	—	—	—
PE9 (port)	TOE03 (ATU-III)	—	—	—	—	—
PE10 (port)	TOE10 (ATU-III)	—	—	—	—	—
PE11 (port)	TOE11 (ATU-III)	—	—	—	—	—
PE12 (port)	TOE12 (ATU-III)	—	—	—	—	—
PE13 (port)	TOE13 (ATU-III)	—	—	—	—	—

Table 23.6 Multiplex Pin List (Port F)

Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)	Function 10 (Related Module)	Function 11 (Related Module)	Function 12 (Related Module)
PF0 (port)	TOD00B (ATU-III)	—	TIF6 (ATU-III)	—	—	—
PF1 (port)	TOD01B (ATU-III)	—	TIF7 (ATU-III)	—	—	—
PF2 (port)	TOD02B (ATU-III)	—	TIF8 (ATU-III)	—	—	—
PF3 (port)	TOD03B (ATU-III)	—	TIF9 (ATU-III)	—	—	—
PF4 (port)	TOD10B (ATU-III)	—	TIF10 (ATU-III)	—	—	—
PF5 (port)	TOD11B (ATU-III)	—	TIF11 (ATU-III)	—	—	—
PF6 (port)	TOD12B (ATU-III)	—	TIF12 (ATU-III)	—	—	—
PF7 (port)	TOD13B (ATU-III)	—	TIF13 (ATU-III)	—	—	—
PF8 (port)	TOD20B (ATU-III)	—	TIF14 (ATU-III)	—	—	—
PF9 (port)	TOD21B (ATU-III)	—	TIF15 (ATU-III)	—	—	—
PF10 (port)	TOD22B (ATU-III)	—	TIF16 (ATU-III)	—	—	—
PF11 (port)	TOD23B (ATU-III)	—	TIF17 (ATU-III)	—	—	—
PF12 (port)	TOD30B (ATU-III)	—	TIF18 (ATU-III)	—	—	—
PF13 (port)	TOD31B (ATU-III)	—	TIF19 (ATU-III)	—	—	—
PF14 (port)	TOD32B (ATU-III)	CTx_B (RCAN-TL1)	TxD_A (SCI)	—	—	FRTxD_B (FlexRay)
PF15 (port)	TOD33B (ATU-III)	CRx_B (RCAN-TL1)	RxD_A (SCI)	—	—	FRRxD_B (FlexRay)

Table 23.7 Multiplex Pin List (Port G)

Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)	Function 10 (Related Module)	Function 11 (Related Module)	Function 12 (Related Module)
PG0 (port)	TOD00A (ATU-III)	SSLA0 (RSPI)	—	—	—	—
PG1 (port)	TOD01A (ATU-III)	SSLA1 (RSPI)	—	—	—	—
PG2 (port)	TOD02A (ATU-III)	SSLA2 (RSPI)	—	—	—	—
PG3 (port)	TOD03A (ATU-III)	SSLA3 (RSPI)	—	—	—	—
PG4 (port)	TOD10A (ATU-III)	SSLA4 (RSPI)	SSLB3 (RSPI)	—	—	—
PG5 (port)	TOD11A (ATU-III)	SSLA5 (RSPI)	SSLC3 (RSPI)	—	—	—
PG6 (port)	TOD12A (ATU-III)	SSLB0 (RSPI)	—	—	—	—
PG7 (port)	TOD13A (ATU-III)	SSLB1 (RSPI)	—	—	—	—
PG8 (port)	TOD20A (ATU-III)	SSLB2 (RSPI)	TIF6 (ATU-III)	—	—	—
PG9 (port)	TOD21A (ATU-III)	SSLC0 (RSPI)	TIF7 (ATU-III)	—	—	—
PG10 (port)	TOD22A (ATU-III)	SSLC1 (RSPI)	TIF8 (ATU-III)	—	—	—
PG11 (port)	TOD23A (ATU-III)	SSLC2 (RSPI)	TIF9 (ATU-III)	—	—	—
PG12 (port)	TOD30A (ATU-III)	SSLA4 (RSPI)	TIF10 (ATU-III)	—	—	—
PG13 (port)	TOD31A (ATU-III)	SSLA5 (RSPI)	TIF11 (ATU-III)	—	—	—
PG14 (port)	TOD32A (ATU-III)	SSLA6 (RSPI)	TIF12 (ATU-III)	—	—	—
PG15 (port)	TOD33A (ATU-III)	SSLA7 (RSPI)	TIF13 (ATU-III)	—	—	—

Table 23.8 Multiplex Pin List (Port H)

Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)	Function 10 (Related Module)	Function 11 (Related Module)	Function 12 (Related Module)
PH0 (port)	—	ADTRG_A (ADC)	TIF0A (ATU-III)	—	—	—
PH1 (port)	—	ADTRG_B (ADC)	TIF1A (ATU-III)	—	—	—
PH2 (port)	—	—	TIF2A (ATU-III)	—	—	—
PH3 (port)	—	—	TIF3 (ATU-III)	—	—	—
PH4 (port)	—	—	TIF4 (ATU-III)	—	—	—
PH5 (port)	—	—	TIF5 (ATU-III)	—	—	—

Table 23.9 Multiplex Pin List (Port J)

Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)	Function 10 (Related Module)	Function 11 (Related Module)	Function 12 (Related Module)
PJ0 (port)	TxD_A (SCI)	CTx_A (RCAN-TL1)	CTx_A & CTx_B (RCAN-TL1)	—	TxD_B (SCI)	FRTxD_A (FlexRay)
PJ1 (port)	RxD_A (SCI)	CRx_A (RCAN-TL1)	CRx_A & CRx_B (RCAN-TL1)	—	RxD_B (SCI)	FRRxD_A (FlexRay)
PJ2 (port)	TxD_A (SCI)	CTx_C (RCAN-TL1)	CTx_A & CTx_B & CTx_C (RCAN-TL1)	—	TxD_B (SCI)	CTx_C & CTx_D (RCAN)
PJ3 (port)	RxD_A (SCI)	CRx_C (RCAN-TL1)	CRx_A & CRx_B & CRx_C (RCAN-TL1)	—	RxD_B (SCI)	CRx_C & CRx_D (RCAN)
PJ4 (port)	SCK_A (SCI)	ADEND_B (ADC)	TIJ0 (ATU-III)	—	—	—
PJ5 (port)	TxD_A (SCI)	—	—	—	—	—
PJ6 (port)	RxD_A (SCI)	—	—	—	—	—
PJ7 (port)	SCK_B (SCI)	ADEND_A (ADC)	TIJ1 (ATU-III)	—	—	FREN_A (FlexRay)
PJ8 (port)	TxD_B (SCI)	—	—	—	—	CTx_D (RCAN)
PJ9 (port)	RxD_B (SCI)	—	—	—	—	CRx_D (RCAN)

Table 23.10 Multiplex Pin List (Port K)

Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)	Function 10 (Related Module)	Function 11 (Related Module)	Function 12 (Related Module)
PK0 (port)	SCK_C (SCI)	RSPCKA (RSPI)	$\overline{\text{UBCTRG}}$ (UBC)	—	—	—
PK1 (port)	TxD_C (SCI)	MOSIA (RSPI)	—	—	—	—
PK2 (port)	RxD_C (SCI)	MISOA (RSPI)	—	—	—	—
PK3 (port)	SCK_D (SCI)	RSPCKB (RSPI)	—	—	—	—
PK4 (port)	TxD_D (SCI)	MOSIB (RSPI)	—	—	—	—
PK5 (port)	RxD_D (SCI)	MISOB (RSPI)	—	—	—	—
PK6 (port)	SCK_E (SCI)	—	—	—	—	—
PK7 (port)	TxD_E (SCI)	—	—	—	—	—
PK8 (port)	RxD_E (SCI)	—	—	—	—	—
PK9 (port)	—	RSPCKC (RSPI)	—	—	—	—
PK10 (port)	—	MOSIC (RSPI)	—	—	—	—
PK11 (port)	—	MISOC (RSPI)	—	—	—	—

Table 23.11 Multiplex Pin List (Port L)

Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)	Function 10 (Related Module)	Function 11 (Related Module)	Function 12 (Related Module)
PL0 (port)	—	$\overline{\text{IRQ0}}$ (INTC)	—	—	—	—
PL1 (port)	TOE20 (ATU-III)	$\overline{\text{IRQ1}}$ (INTC)	$\overline{\text{POD}}$ (port)	—	—	—
PL2 (port)	TOE21 (ATU-III)	$\overline{\text{IRQ2}}$ (INTC)	—	—	—	—
PL3 (port)	TOE22 (ATU-III)	$\overline{\text{IRQ3}}$ (INTC)	—	—	—	—
PL4 (port)	TOE23 (ATU-III)	$\overline{\text{IRQ4}}$ (INTC)	—	—	—	—
PL5 (port)	TOE30 (ATU-III)	$\overline{\text{IRQ5}}$ (INTC)	—	—	—	—
PL6 (port)	TOE31 (ATU-III)	$\overline{\text{IRQ6}}$ (INTC)	—	—	—	—
PL7 (port)	TOE32 (ATU-III)	$\overline{\text{IRQ7}}$ (INTC)	—	—	—	—
PL8 (port)	TOE33 (ATU-III)	—	—	—	—	—

23.1 Register Descriptions

The PFC registers are summarized in table 23.12.

Table 23.12 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port A I/O register	PAIOR	R/W	H'0000	H'FFFE3806	8, 16
Port A control register 4	PACR4	R/W	H'1111*[1] H'1111*[2] H'1111*[3] H'0000*[4]	H'FFFE3810	8, 16, 32
Port A control register 3	PACR3	R/W	H'1111*[1] H'1111*[2] H'1111*[3] H'0000*[4]	H'FFFE3812	8, 16
Port A control register 2	PACR2	R/W	H'1111*[1] H'1111*[2] H'1111*[3] H'0000*[4]	H'FFFE3814	8, 16, 32
Port A control register 1	PACR1	R/W	H'1111*[1] H'1111*[2] H'1111*[3] H'0000*[4]	H'FFFE3816	8, 16
Port B I/O register	PBIOR	R/W	H'0000	H'FFFE3886	8, 16
Port B control register 4	PBCR4	R/W	H'0400*[1] H'0400*[2] H'0400*[3] H'0000*[4]	H'FFFE3890	8, 16, 32
Port B control register 3	PBCR3	R/W	H'0110*[1] H'0110*[2] H'0112*[3] H'0000*[4]	H'FFFE3892	8, 16
Port B control register 2	PBCR2	R/W	H'0122*[1] H'1122*[2] H'1122*[3] H'0000*[4]	H'FFFE3894	8, 16, 32
Port B control register 1	PBCR1	R/W	H'2222*[1] H'2222*[2] H'2222*[3] H'0000*[4]	H'FFFE3896	8, 16

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port C I/O register	PCIOR	R/W	H'0000	H'FFFE3906	8, 16
Port C control register 4	PCCR4	R/W	H'0000*[1] H'1111*[2] H'1111*[3] H'0000*[4]	H'FFFE3910	8, 16, 32
Port C control register 3	PCCR3	R/W	H'0000*[1] H'1111*[2] H'1111*[3] H'0000*[4]	H'FFFE3912	8, 16
Port C control register 2	PCCR2	R/W	H'1111*[1] H'1111*[2] H'1111*[3] H'0000*[4]	H'FFFE3914	8, 16, 32
Port C control register 1	PCCR1	R/W	H'1111*[1] H'1111*[2] H'1111*[3] H'0000*[4]	H'FFFE3916	8, 16
Port D I/O register	PDIOR	R/W	H'0000	H'FFFFC808	8, 16
Port D control register 2	PDCR2	R/W	H'0000	H'FFFFC80C	8, 16, 32
Port D control register 1	PDCR1	R/W	H'0000	H'FFFFC80E	8, 16
Port E I/O register	PEIOR	R/W	H'0000	H'FFFFC818	8, 16
Port E control register 2	PECR2	R/W	H'0000	H'FFFFC81C	8, 16, 32
Port E control register 1	PECR1	R/W	H'0000	H'FFFFC81E	8, 16
Port F I/O register	PFIOR	R/W	H'0000	H'FFFFC82A	8, 16
Port F control register 2	PFCR2	R/W	H'0000	H'FFFFC82C	8, 16, 32
Port F control register 1	PFCR1	R/W	H'0000	H'FFFFC82E	8, 16
Port G I/O register	PGIOR	R/W	H'0000	H'FFFFC83C	8, 16
Port G control register 2	PGCR2	R/W	H'0000	H'FFFFC840	8, 16, 32
Port G control register 1	PGCR1	R/W	H'0000	H'FFFFC842	8, 16
Port H I/O register	PHIOR	R/W	H'0000	H'FFFFC854	8, 16
Port H control register	PHCR	R/W	H'0000	H'FFFFC858	8, 16
Port J I/O register	PJIOR	R/W	H'0000	H'FFFFC86C	8, 16
Port J control register 2	PJCR2	R/W	H'0000	H'FFFFC870	8, 16, 32
Port J control register 1	PJCR1	R/W	H'0000	H'FFFFC872	8, 16

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port K I/O register	PKIOR	R/W	H'0000	H'FFFFC88C	8, 16
Port K control register 2	PKCR2	R/W	H'0000	H'FFFFC890	8, 16, 32
Port K control register 1	PKCR1	R/W	H'0000	H'FFFFC892	8, 16
Port L I/O register	PLIOR	R/W	H'0000	H'FFFFC8A8	8, 16
Port L control register 2	PLCR2	R/W	H'0000	H'FFFFC8AC	8, 16, 32
Port L control register 1	PLCR1	R/W	H'0000	H'FFFFC8AE	8, 16
Port A control register 4A	PACR4A	R/W	H'0000	H'FFFFCA00	8, 16, 32
Port A control register 3A	PACR3A	R/W	H'0000	H'FFFFCA02	8, 16
Port A control register 2A	PACR2A	R/W	H'0000	H'FFFFCA04	8, 16
Port B control register 4A	PBCR4A	R/W	H'0000	H'FFFFCA10	8, 16, 32
Port B control register 3A	PBCR3A	R/W	H'0000	H'FFFFCA12	8, 16
Port B control register 2A	PBCR2A	R/W	H'0000	H'FFFFCA14	8, 16, 32
Port B control register 1A	PBCR1A	R/W	H'0000	H'FFFFCA16	8, 16
Port C control register 4A	PCCR4A	R/W	H'0000	H'FFFFCA20	8, 16, 32
Port C control register 3A	PCCR3A	R/W	H'0000	H'FFFFCA22	8, 16
Port C control register 2A	PCCR2A	R/W	H'0000	H'FFFFCA24	8, 16
Port D control register 2A	PDCR2A	R/W	H'0000	H'FFFFCA34	8, 16
Port E control register 1A	PECR1A	R/W	H'0000	H'FFFFCA46	8, 16
Port F control register 2A	PFCR2A	R/W	H'0000	H'FFFFCA54	8, 16
Port J control register 3A	PJCR3A	R/W	H'0000	H'FFFFCA82	8, 16
Port J control register 2A	PJCR2A	R/W	H'0000	H'FFFFCA84	8, 16, 32
Port J control register 1A	PJCR1A	R/W	H'0000	H'FFFFCA86	8, 16

Notes: 1. On-chip ROM disabled extension mode (area 0: 8 bits)
2. On-chip ROM disabled extension mode (area 0: 16 bits)
3. On-chip ROM enabled extension mode
4. Single-chip mode

23.1.1 Port A I/O Register (PAIOR)

PAIOR is a 16-bit readable/writable register that sets the I/O direction of the port A pins.

PAIOR is enabled only when the port A pins function as general I/O pins. Otherwise, the set values of this register have no effect on the pin status.

PAIOR is initialized to H'0000 either by power-on reset, hardware standby, or power-on reset by the WDT. However, it is not initialized in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PA15 IOR	PA14 IOR	PA13 IOR	PA12 IOR	PA11 IOR	PA10 IOR	PA9 IOR	PA8 IOR	PA7 IOR	PA6 IOR	PA5 IOR	PA4 IOR	PA3 IOR	PA2 IOR	PA1 IOR	PA0 IOR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	PA15IOR to PA0IOR	All 0	R/W	The PA15IOR to PA0IOR bits correspond to the PA15 to PA0 pins respectively (the multiplex pin names other than the port names are omitted from pin names). 0: The corresponding pin is set to input. 1: The corresponding pin is set to output.

23.1.2 Port A Control Registers 1 to 4 (PACR1 to PACR4)

PACR1 to PACR4 are 16-bit readable/writable registers that control the functions of multiplexed pins of port A. PACR1 to PACR4 are initialized to the values shown in table 23.13 either by power-on reset, hardware standby, or power-on reset by the WDT. However, they are not initialized in sleep mode.

Table 23.13 Initial Values of Port A Control Registers

	Initial Values		
Register Name	**On-Chip ROM Disabled Extension Mode**	**On-Chip ROM Enabled Extension Mode**	**Single-Chip Mode**
PACR4	H'1111	H'1111	H'0000
PACR3	H'1111	H'1111	H'0000
PACR2	H'1111	H'1111	H'0000
PACR1	H'1111	H'1111	H'0000

(1) Port A Control Register 4 (PACR4)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	PA15 MD	-	-	-	PA14 MD	-	-	-	PA13 MD	-	-	-	PA12 MD
Initial value:	0	0	0	0/1*	0	0	0	0/1*	0	0	0	0/1*	0	0	0	0/1*
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Note: * The initial value depends on the operating mode of the LSI.

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	PA15MD	0/1*	R/W	PA15 Mode Controls the function of the PA15/A15 pin. • On-chip ROM disabled extension mode 0: A15 output (BSC) 1: A15 output (BSC) (initial value) • On-chip ROM enabled extension mode 0: PA15 input/output (port) 1: A15 output (BSC) (initial value) • Single-chip mode 0: PA15 input/output (port) (initial value) 1: PA15 input/output (port)
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
8	PA14MD	0/1*	R/W	PA14 Mode Controls the function of the PA14/A14 pin. • On-chip ROM disabled extension mode 0: A14 output (BSC) 1: A14 output (BSC) (initial value) • On-chip ROM enabled extension mode 0: PA14 input/output (port) 1: A14 output (BSC) (initial value) • Single-chip mode 0: PA14 input/output (port) (initial value) 1: PA14 input/output (port)
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	PA13MD	0/1*	R/W	PA13 Mode Controls the function of the PA13/A13 pin. • On-chip ROM disabled extension mode 0: A13 output (BSC) 1: A13 output (BSC) (initial value) • On-chip ROM enabled extension mode 0: PA13 input/output (port) 1: A13 output (BSC) (initial value) • Single-chip mode 0: PA13 input/output (port) (initial value) 1: PA13 input/output (port)
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
0	PA12MD	0/1*	R/W	PA12 Mode Controls the function of the PA12/A12 pin. • On-chip ROM disabled extension mode 0: A12 output (BSC) 1: A12 output (BSC) (initial value) • On-chip ROM enabled extension mode 0: PA12 input/output (port) 1: A12 output (BSC) (initial value) • Single-chip mode 0: PA12 input/output (port) (initial value) 1: PA12 input/output (port)

Note: * The initial value depends on the operating mode of the LSI.

(2) Port A Control Register 3 (PACR3)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	PA11 MD	-	-	-	PA10 MD	-	-	-	PA9 MD	-	-	-	PA8 MD
Initial value:	0	0	0	0/1*	0	0	0	0/1*	0	0	0	0/1*	0	0	0	0/1*
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Note: * The initial value depends on the operating mode of the LSI.

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	PA11MD	0/1*	R/W	PA11 Mode Controls the function of the PA11/A11 pin. • On-chip ROM disabled extension mode 0: A11 output (BSC) 1: A11 output (BSC) (initial value) • On-chip ROM enabled extension mode 0: PA11 input/output (port) 1: A11 output (BSC) (initial value) • Single-chip mode 0: PA11 input/output (port) (initial value) 1: PA11 input/output (port)
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
8	PA10MD	0/1*	R/W	PA10 Mode Controls the function of the PA10/A10 pin. • On-chip ROM disabled extension mode 0: A10 output (BSC) 1: A10 output (BSC) (initial value) • On-chip ROM enabled extension mode 0: PA10 input/output (port) 1: A10 output (BSC) (initial value) • Single-chip mode 0: PA10 input/output (port) (initial value) 1: PA10 input/output (port)
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	PA9MD	0/1*	R/W	PA9 Mode Controls the function of the PA9/A9 pin. • On-chip ROM disabled extension mode 0: A9 output (BSC) 1: A9 output (BSC) (initial value) • On-chip ROM enabled extension mode 0: PA9 input/output (port) 1: A9 output (BSC) (initial value) • Single-chip mode 0: PA9 input/output (port) (initial value) 1: PA9 input/output (port)
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
0	PA8MD	0/1*	R/W	PA8 Mode Controls the function of the PA8/A8 pin. • On-chip ROM disabled extension mode 0: A8 output (BSC) 1: A8 output (BSC) (initial value) • On-chip ROM enabled extension mode 0: PA8 input/output (port) 1: A8 output (BSC) (initial value) • Single-chip mode 0: PA8 input/output (port) (initial value) 1: PA8 input/output (port)

Note: * The initial value depends on the operating mode of the LSI.

(3) Port A Control Register 2 (PACR2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	PA7 MD	-	-	-	PA6 MD	-	-	-	PA5 MD	-	-	-	PA4 MD
Initial value:	0	0	0	0/1*	0	0	0	0/1*	0	0	0	0/1*	0	0	0	0/1*
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Note: * The initial value depends on the operating mode of the LSI.

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	PA7MD	0/1*	R/W	PA7 Mode Controls the function of the PA7/A7 pin. • On-chip ROM disabled extension mode 0: A7 output (BSC) 1: A7 output (BSC) (initial value) • On-chip ROM enabled extension mode 0: PA7 input/output (port) 1: A7 output (BSC) (initial value) • Single-chip mode 0: PA7 input/output (port) (initial value) 1: PA7 input/output (port)
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
8	PA6MD	0/1*	R/W	PA6 Mode Controls the function of the PA6/A6 pin. • On-chip ROM disabled extension mode 0: A6 output (BSC) 1: A6 output (BSC) (initial value) • On-chip ROM enabled extension mode 0: PA6 input/output (port) 1: A6 output (BSC) (initial value) • Single-chip mode 0: PA6 input/output (port) (initial value) 1: PA6 input/output (port)
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	PA5MD	0/1*	R/W	PA5 Mode Controls the function of the PA5/A5 pin. • On-chip ROM disabled extension mode 0: A5 output (BSC) 1: A5 output (BSC) (initial value) • On-chip ROM enabled extension mode 0: PA5 input/output (port) 1: A5 output (BSC) (initial value) • Single-chip mode 0: PA5 input/output (port) (initial value) 1: PA5 input/output (port)
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
0	PA4MD	0/1*	R/W	PA4 Mode Controls the function of the PA4/A4 pin. • On-chip ROM disabled extension mode 0: A4 output (BSC) 1: A4 output (BSC) (initial value) • On-chip ROM enabled extension mode 0: PA4 input/output (port) 1: A4 output (BSC) (initial value) • Single-chip mode 0: PA4 input/output (port) (initial value) 1: PA4 input/output (port)

Note: * The initial value depends on the operating mode of the LSI.

(4) Port A Control Register 1 (PACR1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	PA3 MD	-	-	-	PA2 MD	-	-	-	PA1 MD	-	-	-	PA0 MD
Initial value:	0	0	0	0/1*	0	0	0	0/1*	0	0	0	0/1*	0	0	0	0/1*
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Note: * The initial value depends on the operating mode of the LSI.

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	PA3MD	0/1*	R/W	PA3 Mode Controls the function of the PA3/A3 pin. • On-chip ROM disabled extension mode 0: A3 output (BSC) 1: A3 output (BSC) (initial value) • On-chip ROM enabled extension mode 0: PA3 input/output (port) 1: A3 output (BSC) (initial value) • Single-chip mode 0: PA3 input/output (port) (initial value) 1: PA3 input/output (port)
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
8	PA2MD	0/1*	R/W	PA2 Mode Controls the function of the PA2/A2 pin. • On-chip ROM disabled extension mode 0: A2 output (BSC) 1: A2 output (BSC) (initial value) • On-chip ROM enabled extension mode 0: PA2 input/output (port) 1: A2 output (BSC) (initial value) • Single-chip mode 0: PA2 input/output (port) (initial value) 1: PA2 input/output (port)
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	PA1MD	0/1*	R/W	PA1 Mode Controls the function of the PA1/A1 pin. • On-chip ROM disabled extension mode 0: A1 output (BSC) 1: A1 output (BSC) (initial value) • On-chip ROM enabled extension mode 0: PA1 input/output (port) 1: A1 output (BSC) (initial value) • Single-chip mode 0: PA1 input/output (port) (initial value) 1: PA1 input/output (port)
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
0	PA0MD	0/1*	R/W	PA0 Mode Controls the function of the PA0/A0 pin. • On-chip ROM disabled extension mode 0: A0 output (BSC) 1: A0 output (BSC) (initial value) • On-chip ROM enabled extension mode 0: PA0 input/output (port) 1: A0 output (BSC) (initial value) • Single-chip mode 0: PA0 input/output (port) (initial value) 1: PA0 input/output (port)

Note: * The initial value depends on the operating mode of the LSI.

23.1.3 Port A Control Registers 2A to 4A (PACR2A to PACR4A)

PACR2A to PACR4A are 16-bit readable/writable registers that control the functions of multiplexed pins of port A. PACR2A to PACR4A are initialized to H'0000 either by power-on reset, hardware standby, or power-on reset by the WDT. However, they are not initialized in sleep mode.

(1) Port A Control Register 4A (PACR4A)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	PA15 AMD	-	-	-	PA14 AMD	-	-	-	PA13 AMD	-	-	-	PA12 AMD
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	PA15AMD	0	R/W	PA15 Mode Controls the function of the PA15/A15/TIA03 pin. • On-chip ROM disabled extension mode 0: A15 output (BSC) 1: A15 output (BSC) • On-chip ROM enabled extension mode 0: PA15 input/output (port) 1: TIA03 input (ATU-III) • Single-chip mode 0: PA15 input/output (port) (initial value) 1: TIA03 input (ATU-III) Note: This setting is valid only when PA15MD = 0.
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
8	PA14AMD	0	R/W	PA14 Mode Controls the function of the PA14/A14/TIA02 pin. • On-chip ROM disabled extension mode 0: A14 output (BSC) 1: A14 output (BSC) • On-chip ROM enabled extension mode 0: PA14 input/output (port) 1: TIA02 input (ATU-III) • Single-chip mode 0: PA14 input/output (port) (initial value) 1: TIA02 input (ATU-III) Note: This setting is valid only when PA14MD = 0.
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	PA13AMD	0	R/W	PA13 Mode Controls the function of the PA13/A13/TIA01 pin. • On-chip ROM disabled extension mode 0: A13 output (BSC) 1: A13 output (BSC) • On-chip ROM enabled extension mode 0: PA13 input/output (port) 1: TIA01 input (ATU-III) • Single-chip mode 0: PA13 input/output (port) (initial value) 1: TIA01 input (ATU-III) Note: This setting is valid only when PA13MD = 0.
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
0	PA12AMD	0	R/W	PA12 Mode Controls the function of the PA12/A12/TIA00 pin. • On-chip ROM disabled extension mode 0: A12 output (BSC) 1: A12 output (BSC) • On-chip ROM enabled extension mode 0: PA12 input/output (port) 1: TIA00 input (ATU-III) • Single-chip mode 0: PA12 input/output (port) (initial value) 1: TIA00 input (ATU-III) Note: This setting is valid only when PA12MD = 0.

(2) Port A Control Register 3A (PACR3A)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PA11AMD[1:0]		-	-	PA10AMD[1:0]		-	-	PA9AMD[1:0]		-	-	-	PA8A MD
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13, 12	PA11AMD [1:0]	00	R/W	PA11 Mode Controls the function of the PA11/A11/TOD13B/TIF2B pin. • On-chip ROM disabled extension mode 00: A11 output (BSC) 01: A11 output (BSC) 10: A11 output (BSC) 11: Setting prohibited • On-chip ROM enabled extension mode 00: PA11 input/output (port) 01: TOD13B output (ATU-III) 10: TIF2B input (ATU-III) 11: Setting prohibited • Single-chip mode 0: PA11 input/output (port) (initial value) 01: TOD13B output (ATU-III) 10: TIF2B input (ATU-III) 11: Setting prohibited Note: This setting is valid only when PA11MD = 0.
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
9, 8	PA10AMD [1:0]	00	R/W	PA10 Mode Controls the function of the PA10/A10/TOD12B/TIF1B pin. • On-chip ROM disabled extension mode 00: A10 output (BSC) 01: A10 output (BSC) 10: A10 output (BSC) 11: Setting prohibited • On-chip ROM enabled extension mode 00: PA10 input/output (port) 01: TOD12B output (ATU-III) 10: TIF1B input (ATU-III) 11: Setting prohibited • Single-chip mode 00: PA10 input/output (port) (initial value) 01: TOD12B output (ATU-III) 10: TIF1B input (ATU-III) 11: Setting prohibited Note: This setting is valid only when PA10MD = 0.
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
5, 4	PA9AMD [1:0]	00	R/W	PA9 Mode Controls the function of the PA9/A9/TOD11B/TIF0B pin. • On-chip ROM disabled extension mode 00: A9 output (BSC) 01: A9 output (BSC) 10: A9 output (BSC) 11: Setting prohibited • On-chip ROM enabled extension mode 00: PA9 input/output (port) 01: TOD11B output (ATU-III) 10: TIF0B input (ATU-III) 11: Setting prohibited • Single-chip mode 00: PA9 input/output (port) (initial value) 01: TOD11B output (ATU-III) 10: TIF0B input (ATU-III) 11: Setting prohibited Note: This setting is valid only when PA9MD = 0.
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	PA8AMD	0	R/W	PA8 Mode Controls the function of the PA8/A8/TOD10B pin. • On-chip ROM disabled extension mode 0: A8 output (BSC) 1: A8 output (BSC) • On-chip ROM enabled extension mode 0: PA8 input/output (port) 1: TOD10B output (ATU-III) • Single-chip mode 0: PA8 input/output (port) (initial value) 1: TOD10B output (ATU-III) Note: This setting is valid only when PA8MD = 0.

(3) Port A Control Register 2A (PACR2A)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PA7AMD[1:0]		-	-	PA6AMD[1:0]		-	-	PA5AMD[1:0]		-	-	-	PA4 AMD
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13, 12	PA7AMD [1:0]	00	R/W	PA7 Mode Controls the function of the PA7/A7/TOD03B/TIF2A pin. • On-chip ROM disabled extension mode 00: A7 output (BSC) 01: A7 output (BSC) 10: A7 output (BSC) 11: Setting prohibited • On-chip ROM enabled extension mode 00: PA7 input/output (port) 01: TOD03B output (ATU-III) 10: TIF2A input (ATU-III) 11: Setting prohibited • Single-chip mode 00: PA7 input/output (port) (initial value) 01: TOD03B output (ATU-III) 10: TIF2A input (ATU-III) 11: Setting prohibited Note: This setting is valid only when PA7MD = 0.
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
9, 8	PA6AMD [1:0]	00	R/W	PA6 Mode Controls the function of the PA6/A6/TOD02B/TIF1A pin. • On-chip ROM disabled extension mode 00: A6 output (BSC) 01: A6 output (BSC) 10: A6 output (BSC) 11: Setting prohibited • On-chip ROM enabled extension mode 00: PA6 input/output (port) 01: TOD02B output (ATU-III) 10: TIF1A input (ATU-III) 11: Setting prohibited • Single-chip mode 00: PA6 input/output (port) (initial value) 01: TOD02B output (ATU-III) 10: TIF1A input (ATU-III) 11: Setting prohibited Note: This setting is valid only when PA6MD = 0.
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
5, 4	PA5AMD [1:0]	00	R/W	PA5 Mode Controls the function of the PA5/A5/TOD01B/TIF0A pin. • On-chip ROM disabled extension mode 00: A5 output (BSC) 01: A5 output (BSC) 10: A5 output (BSC) 11: Setting prohibited • On-chip ROM enabled extension mode 00: PA5 input/output (port) 01: TOD01B output (ATU-III) 10: TIF0A output (ATU-III) 11: Setting prohibited • Single-chip mode 00: PA5 input/output (port) (initial value) 01: TOD00B output (ATU-III) 10: TIF0A output (ATU-III) 11: Setting prohibited Note: This setting is valid only when PA5MD = 0.
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	PA4AMD	0	R/W	PA4 Mode Controls the function of the PA4/A4/TOD00B pin. • On-chip ROM disabled extension mode 0: A4 output (BSC) 1: A4 output (BSC) • On-chip ROM enabled extension mode 0: PA4 input/output (port) 1: TOD00B output (ATU-III) • Single-chip mode 0: PA4 input/output (port) (initial value) 1: TOD00B output (ATU-III) Note: This setting is valid only when PA4MD = 0.

23.1.4 Port B I/O Register (PBIOR)

PBIOR is a 16-bit readable/writable register that sets the I/O direction of the port B pins.

PBIOR is enabled only when the port B pins function as general I/O pins. Otherwise, the set values of this register have no effect on the pin status.

PBIOR is initialized to H'0000 either by power-on reset, hardware standby, or power-on reset by the WDT. However, it is not initialized in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PB14 IOR	PB13 IOR	PB12 IOR	PB11 IOR	PB10 IOR	PB9 IOR	PB8 IOR	PB7 IOR	PB6 IOR	PB5 IOR	PB4 IOR	PB3 IOR	PB2 IOR	PB1 IOR	PB0 IOR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 0	PB14IOR to PB0IOR	All 0	R/W	The PB14IOR to PB0IOR bits correspond to the PB14 to PB0 pins respectively (the multiplex pin names other than the port names are omitted from pin names). 0: The corresponding pin is set to input. 1: The corresponding pin is set to output.

23.1.5 Port B Control Registers 1 to 4 (PBCR1 to PBCR4)

PBCR1 to PBCR4 are 16-bit readable/writable registers that control the functions of multiplexed pins of port B.

PBCR1 to PBCR4 are initialized to the values shown in table 23.14 either by power-on reset, hardware standby, or power-on reset by the WDT. However, they are not initialized in sleep mode.

Table 23.14 Initial Values of Port B Control Registers

	Initial Values			
	On-Chip ROM Disabled Extension Mode		On-Chip ROM Enabled Extension Mode	
Register Name	**Area 0: 8 Bits**	**Area 0: 16 Bits**		**Single-Chip Mode**
PBCR4	H'0400	H'0400	H'0400	H'0000
PBCR3	H'0110	H'0110	H'0112	H'0000
PBCR2	H'0122	H'1122	H'1122	H'0000
PBCR1	H'2222	H'2222	H'2222	H'0000

(1) Port B Control Register 4 (PBCR4)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	PB14MD[2:0]			-	-	PB13MD[1:0]		-	-	PB12MD[1:0]	
Initial value:	0	0	0	0	0	0/1*	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

Note: * The initial value depends on the operating mode of the LSI.

Bit	Bit Name	Initial Value	R/W	Description
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
10 to 8	PB14MD [2:0]	000/100*	R/W	PB14 Mode Control the function of the PB14/RD/$\overline{\text{WR}}$ pin. • On-chip ROM disabled extension mode 000: RD/$\overline{\text{WR}}$ output (BSC) 001: Setting prohibited 010: Setting prohibited 011: Setting prohibited 100: RD/$\overline{\text{WR}}$ output (BSC) (initial value) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited • On-chip ROM enabled extension mode 000: PB14 input/output (port) 001: Setting prohibited 010: Setting prohibited 011: Setting prohibited 100: RD/$\overline{\text{WR}}$ output (BSC) (initial value) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited • Single-chip mode 000: PB14 input/output (port) (initial value) 001: Setting prohibited 010: Setting prohibited 011: Setting prohibited 100: PB14 input/output (port) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
5, 4	PB13MD [1:0]	00	R/W	PB13 Mode Control the function of the PB13/$\overline{\text{CS3}}$/RSPCKB pin. • Extension mode 00: PB13 input/output (port) (initial value) 01: Setting prohibited 10: $\overline{\text{CS3}}$ output (BSC) 11: RSPCKB input/output (RSPI) • Single-chip mode 00: PB13 input/output (port) (initial value) 01: Setting prohibited 10: PB13 input/output (port) 11: RSPCKB input/output (RSPI)
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	PB12MD [1:0]	00	R/W	PA12 Mode Control the function of the PB12/$\overline{\text{CS2}}$/RSPCKA pin. • Extension mode 00: PB12 input/output (port) (initial value) 01: Setting prohibited 10: $\overline{\text{CS2}}$ output (BSC) 11: RSPCKA input/output (RSPI) • Single-chip mode 00: PB12 input/output (port) (initial value) 01: Setting prohibited 10: PB12 input/output (port) 11: RSPCKA input/output (RSPI)

Note: * The initial value depends on the operating mode of the LSI.

(2) Port B Control Register 3 (PBCR3)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PB11MD[1:0]		-	-	-	PB10 MD	-	-	-	PB9 MD	-	-	PB8MD[1:0]	
Initial value:	0	0	0	0	0	0	0	0/1*	0	0	0	0/1*	0	0	0/1*	0
R/W:	R	R	R/W	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R/W	R/W

Note: * The initial value depends on the operating mode of the LSI.

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13, 12	PB11MD [1:0]	00	R/W	PB11 Mode Control the function of the PB11/$\overline{\text{CS1}}$/TOE21 pin. • Extension mode 00: PB11 input/output (port) (initial value) 01: Setting prohibited 10: $\overline{\text{CS1}}$ output (BSC) 11: TOE21 output (ATU-III) • Single-chip mode 00: PB11 input/output (port) (initial value) 01: Setting prohibited 10: PB11 input/output (port) 11: TOE21 output (ATU-III)
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
8	PB10MD	0/1*	R/W	PB10 Mode Controls the function of the PB10/$\overline{\text{CS0}}$ pin. • On-chip ROM disabled extension mode 0: $\overline{\text{CS0}}$ output (BSC) 1: $\overline{\text{CS0}}$ output (BSC) (initial value) • On-chip ROM enabled extension mode 0: PB10 input/output (port) 1: $\overline{\text{CS0}}$ output (BSC) (initial value) • Single-chip mode 0: PB10 input/output (port) (initial value) 1: PB10 input/output (port)
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	PB9MD	0/1*	R/W	PB9 Mode Controls the function of the PB9/$\overline{\text{RD}}$ pin. • On-chip ROM disabled extension mode 0: $\overline{\text{RD}}$ output (BSC) 1: $\overline{\text{RD}}$ output (BSC) (initial value) • On-chip ROM enabled extension mode 0: PB9 input/output (port) 1: $\overline{\text{RD}}$ output (BSC) (initial value) • Single-chip mode 0: PB9 input/output (port) (initial value) 1: PB9 input/output (port)
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
1, 0	PB8MD [1:0]	00/10*	R/W	PB8 Mode Control the function of the PB8/$\overline{\text{WAIT}}$/TOE20 pin. • On-chip ROM disabled extension mode 00: $\overline{\text{WAIT}}$ input (BSC) (initial value) 01: Setting prohibited 10: $\overline{\text{WAIT}}$ input (BSC) 11: $\overline{\text{WAIT}}$ input (BSC) • On-chip ROM enabled extension mode 00: PB8 input/output (port) 01: Setting prohibited 10: $\overline{\text{WAIT}}$ input (BSC) (initial value) 11: TOE20 output (ATU-III) • Single-chip mode 00: PB8 input/output (port) (initial value) 01: Setting prohibited 10: PB8 input/output (port) (BSC) 11: TOE20 output (ATU-III)

Note: * The initial value depends on the operating mode of the LSI.

(3) Port B Control Register 2 (PBCR2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	PB7 MD	-	-	-	PB6 MD	-	PB5MD[2:0]			-	PB4MD[2:0]		
Initial value:	0	0	0	0/1*	0	0	0	0/1*	0	0	0/1*	0	0	0	0/1*	0
R/W:	R	R	R	R/W	R	R	R	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Note: * The initial value depends on the operating mode of the LSI.

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	PB7MD	0/1*	R/W	PB7 Mode Controls the function of the PB7/$\overline{\text{WE1}}$ pin. • On-chip ROM disabled extension mode (area 0: 8 bits) 0: PB7 input/output (port) (initial value) 1: $\overline{\text{WE1}}$ output (BSC) • On-chip ROM disabled extension mode (area 0: 16 bits) 0: $\overline{\text{WE1}}$ output (BSC) 1: $\overline{\text{WE1}}$ output (BSC) (initial value) • On-chip ROM enabled extension mode 0: PB7 input/output (port) 1: $\overline{\text{WE1}}$ output (BSC) (initial value) • Single-chip mode 0: PB7 input/output (port) (initial value) 1: PB7 input/output (port)
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
8	PB6MD	0/1*	R/W	PB6 Mode Controls the function of the PB6/$\overline{\text{WE0}}$ pin. • On-chip ROM disabled extension mode 0: $\overline{\text{WE0}}$ output (BSC) 1: $\overline{\text{WE0}}$ output (BSC) (initial value) • On-chip ROM enabled extension mode 0: PB6 input/output (port) 1: $\overline{\text{WE0}}$ output (BSC) (initial value) • Single-chip mode 0: PB6 input/output (port) (initial value) 1: PB6 input/output (port)
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
6 to 4	PB5MD [2:0]	000/010*	R/W	PB5 Mode Control the function of the PB5/A21/CRx_B/TIF7 pin. • On-chip ROM disabled extension mode 000: A21 output (address) 001: Setting prohibited 010: A21 output (address) (initial value) 011: A21 output (address) 100: A21 output (address) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited • On-chip ROM enabled extension mode 000: PB5 input/output (port) 001: Setting prohibited 010: A21 output (address) (initial value) 011: CRx_B input (RCAN-TL1) 100: TIF7 input (ATU-III) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited • Single-chip mode 000: PB5 input/output (port) (initial value) 001: Setting prohibited 010: PB5 input/output (port) 011: CRx_B input (RCAN-TL1) 100: TIF7 input (ATU-III) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
2 to 0	PB4MD [2:0]	000/010*	R/W	PB4 Mode Control the function of the PB4/A20/CTx_B/TIF6 pin. • On-chip ROM disabled extension mode 000: A20 output (address) 001: Setting prohibited 010: A20 output (address) (initial value) 011: A20 output (address) 100: A20 output (address) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited • On-chip ROM enabled extension mode 000: PB4 input/output (port) 001: Setting prohibited 010: A20 output (address) (initial value) 011: CTx_B output (RCAN-TL1) 100: TIF6 input (ATU-III) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited • Single-chip mode 000: PB4 input/output (port) (initial value) 001: Setting prohibited 010: PB4 input/output (port) 011: CTx_B output (RCAN-TL1) 100: TIF6 input (ATU-III) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

Note: * The initial value depends on the operating mode of the LSI.

(4) Port B Control Register 1 (PBCR1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PB3MD[1:0]		-	-	PB2MD[1:0]		-	-	PB1MD[1:0]		-	-	PB0MD[1:0]	
Initial value:	0	0	0/1*	0	0	0	0/1*	0	0	0	0/1*	0	0	0	0/1*	0
R/W:	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

Note: * The initial value depends on the operating mode of the LSI.

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13, 12	PB3MD [1:0]	00/10*	R/W	PB3 Mode Control the function of the PB3/A19/MISOB pin. • On-chip ROM disabled extension mode 00: A19 output (address) 01: Setting prohibited 10: A19 output (address) (initial value) 11: A19 output (address) • On-chip ROM enabled extension mode 00: PB3 input/output (port) 01: Setting prohibited 10: A19 output (address) (initial value) 11: MISOB input/output (RSPI) • Single-chip mode 00: PB3 input/output (port) (initial value) 01: Setting prohibited 10: PB3 input/output (port) 11: MISOB input/output (RSPI)
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
9, 8	PB2MD [1:0]	00/10*	R/W	PB2 Mode Control the function of the PB2/A18/MOSIB pin. • On-chip ROM disabled extension mode 00: A18 output (address) 01: Setting prohibited 10: A18 output (address) (initial value) 11: A18 output (address) • On-chip ROM enabled extension mode 00: PB2 input/output (port) 01: Setting prohibited 10: A18 output (address) (initial value) 11: MOSIB input/output (RSPI) • Single-chip mode 00: PB2 input/output (port) (initial value) 01: Setting prohibited 10: PB2 input/output (port) 11: MOSIB input/output (RSPI)
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
5, 4	PB1MD [1:0]	00/10*	R/W	PB1 Mode Control the function of the PB1/A17/MISOA pin. • On-chip ROM disabled extension mode 00: A17 output (address) 01: Setting prohibited 10: A17 output (address) (initial value) 11: A17 output (address) • On-chip ROM enabled extension mode 00: PB1 input/output (port) 01: Setting prohibited 10: A17 output (address) (initial value) 11: MISOA input/output (RSPI) • Single-chip mode 00: PB1 input/output (port) (initial value) 01: Setting prohibited 10: PB1 input/output (port) 11: MISOA input/output (RSPI)
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
1, 0	PB0MD [1:0]	00/10*	R/W	PB0 Mode Control the function of the PB0/A16/MOSIA pin. • On-chip ROM disabled extension mode 00: A16 output (address) 01: Setting prohibited 10: A16 output (address) (initial value) 11: A16 output (address) • On-chip ROM enabled extension mode 00: PB0 input/output (port) 01: Setting prohibited 10: A16 output (address) (initial value) 11: MOSIA input/output (RSPI) • Single-chip mode 00: PB0 input/output (port) (initial value) 01: Setting prohibited 10: PB0 input/output (port) 11: MOSIA input/output (RSPI)

Note: * The initial value depends on the operating mode of the LSI.

23.1.6 Port B Control Registers 1A to 4A (PBCR1A to PBCR4A)

PBCR1A to PBCR4A are 16-bit readable/writable registers that control the functions of multiplexed pins of port B.

PBCR1A to PBCR4A are initialized to H'0000 either by power-on reset, hardware standby, or power-on reset by the WDT. However, they are not initialized in sleep mode.

(1) Port B Control Register 4A (PBCR4A)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	PB13AMD	-	-	PB12AMD[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	PB13AMD	0	R/W	PB13 Mode Control the function of the PB13/RSPCKB/TIF22 pin. 0: PB13 input/output (port) (initial value) 1: TIF22 input (ATU-III) Note: This setting is valid only when PB13MD[1:0] = 00.
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	PB12AMD [1:0]	00	R/W	PB12 Mode Control the function of the PB12/FREN_B pin. 00: PB12 input/output (port) (initial value) 01: Setting prohibited 10: Setting prohibited 11: FREN_B output (FlexRay) Note: This setting is valid only when PB12MD[1:0] = 00.

(2) Port B Control Register 3A (PBCR3A)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PB11AMD[1:0]		-	-	PB10AMD[1:0]		-	-	PB9AMD[1:0]		-	-	PB8AMD[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13, 12	PB11AMD [1:0]	00	R/W	PB11 Mode Control the function of the PB11/TIF21/SSLB2 pin. 00: PB11 input/output (port) (initial value) 01: TIF21 input (ATU-III) 10: Setting prohibited 11: SSLB2 output (RSPI) Note: This setting is valid only when PB11MD[1:0] = 00.
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
9, 8	PB10AMD [1:0]	00	R/W	PB10 Mode Controls the function of the PB10/$\overline{\text{CS0}}$/SSLB1 pin. • On-chip ROM disabled extension mode 00: $\overline{\text{CS0}}$ output (BSC) 01: Setting prohibited 10: Setting prohibited 11: $\overline{\text{CS0}}$ output (BSC) • On-chip ROM enabled extension mode 00: PB10 input/output (port) 01: Setting prohibited 10: Setting prohibited 11: SSLB1 output (RSPI) • Single-chip mode 00: PB10 input/output (port) (initial value) 01: Setting prohibited 10: Setting prohibited 11: SSLB1 output (RSPI) Note: This setting is valid only when PB10MD = 0.
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
5, 4	PB9AMD [1:0]	00	R/W	PB9 Mode Controls the function of the PB9/$\overline{\text{RD}}$/SSLB0 pin. • On-chip ROM disabled extension mode 00: $\overline{\text{RD}}$ output (BSC) 01: Setting prohibited 10: Setting prohibited 11: $\overline{\text{RD}}$ output (BSC) • On-chip ROM enabled extension mode 00: PB9 input/output (port) 01: Setting prohibited 10: Setting prohibited 11: SSLB0 input/output (RSPI) • Single-chip mode 00: PB9 input/output (port) (initial value) 01: Setting prohibited 10: Setting prohibited 11: SSLB0 input/output (RSPI) Note: This setting is valid only when PB9MD = 0.
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
1, 0	PB8AMD [1:0]	00	R/W	PB8 Mode Control the function of the PB8/$\overline{\text{WAIT}}$/TIF20/RxD_A pin. • On-chip ROM disabled extension mode 00: $\overline{\text{WAIT}}$ input (BSC) (initial value) 01: $\overline{\text{WAIT}}$ input (BSC) 10: Setting prohibited 11: $\overline{\text{WAIT}}$ input (BSC) • On-chip ROM enabled extension mode 00: PB8 input/output (port) 01: TIF20 input (ATU-III) 10: Setting prohibited 11: RxD_A input (SCI) • Single-chip mode 00: PB8 input/output (port) (initial value) 01: TIF20 input (ATU-III) 10: Setting prohibited 11: RxD_A input (SCI) Note: This setting is valid only when PB8MD[1:0] = 00.

(3) Port B Control Register 2A (PBCR2A)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PB7AMD[1:0]		-	-	PB6AMD[1:0]		-	-	PB5AMD[1:0]		-	-	PB4AMD[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13, 12	PB7AMD [1:0]	00	R/W	PB7 Mode Controls the function of the PB7/$\overline{\text{WE1}}$/TxD_A pin. • On-chip ROM disabled extension mode (area 0: 8 bits) 00: PB7 input/output (port) (initial value) 01: Setting prohibited 10: Setting prohibited 11: TxD_A output (SCI) • On-chip ROM disabled extension mode (area 0: 16 bits) 00: $\overline{\text{WE1}}$ output (BSC) 01: Setting prohibited 10: Setting prohibited 11: $\overline{\text{WE1}}$ output (BSC) • On-chip ROM enabled extension mode 00: PB7 input/output (port) 01: Setting prohibited 10: Setting prohibited 11: TxD_A output (SCI) • Single-chip mode 00: PB7 input/output (port) (initial value) 01: Setting prohibited 10: Setting prohibited 11: TxD_A output (SCI) Note: This setting is valid only when PB7MD = 0.

Bit	Bit Name	Initial Value	R/W	Description
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	PB6AMD [1:0]	00	R/W	PB6 Mode Controls the function of the PB6/$\overline{\text{WE0}}$/SCK_B pin. • On-chip ROM disabled extension mode 00: $\overline{\text{WE0}}$ output (BSC) 01: Setting prohibited 10: Setting prohibited 11: $\overline{\text{WE0}}$ output (BSC) • On-chip ROM enabled extension mode 00: PB6 input/output (port) 01: Setting prohibited 10: Setting prohibited 11: SCK_B input/output (SCI) • Single-chip mode 00: PB6 input/output (port) (initial value) 01: Setting prohibited 10: Setting prohibited 11: SCK_B input/output (SCI) Note: This setting is valid only when PB6MD = 0.
7, 6	—	All 0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
5, 4	PB5AMD [1:0]	00	R/W	PB5 Mode Control the function of the PB5/A21/TIF27/RxD_B pin. • On-chip ROM disabled extension mode 00: A21 output (address) 01: A21 output (address) 10: Setting prohibited 11: A21 output (address) • On-chip ROM enabled extension mode 00: PB5 input/output (port) 01: TIF27 input (ATU-III) 10: Setting prohibited 11: RxD_B input (SCI) • Single-chip mode 00: PB5 input/output (port) (initial value) 01: TIF27 input (ATU-III) 10: Setting prohibited 11: RxD_B input (SCI) Note: This setting is valid only when PB5MD[2:0] = 000.
3, 2	—	All 0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
1, 0	PB4AMD [1:0]	00	R/W	PB4 Mode Control the function of the PB4/A20/TIF26/TxD_B pin. • On-chip ROM disabled extension mode 00: A20 output (address) 01: A20 output (address) 10: Setting prohibited 11: A20 output (address) • On-chip ROM enabled extension mode 00: PB4 input/output (port) 01: TIF26 input (ATU-III) 10: Setting prohibited 11: TxD_B output (SCI) • Single-chip mode 00: PB4 input/output (port) (initial value) 01: TIF26 input (ATU-III) 10: Setting prohibited 11: TxD_B output (SCI) Note: This setting is valid only when PB4MD[2:0] = 000.

(4) Port B Control Register 1A (PBCR1A)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PB3AMD[1:0]		-	-	PB2AMD[1:0]		-	-	PB1AMD[1:0]		-	-	PB0AMD[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13, 12	PB3AMD [1:0]	00	R/W	PB3 Mode Control the function of the PB3/A19/TIA05/SSLA7 pin. • On-chip ROM disabled extension mode 00: A19 output (address) 01: A19 output (address) 10: Setting prohibited 11: A19 output (address) • On-chip ROM enabled extension mode 00: PB3 input/output (port) 01: TIA05 input (ATU-III) 10: Setting prohibited 11: SSLA7 input (RSPI) • Single-chip mode 00: PB3 input/output (port) (initial value) 01: TIA05 input (ATU-III) 10: Setting prohibited 11: SSLA7 input (RSPI) Note: This setting is valid only when PB3MD[1:0] = 00.
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
9, 8	PB2AMD [1:0]	00	R/W	PB2 Mode Control the function of the PB2/A18/TIA04/SSLA6 pin. • On-chip ROM disabled extension mode 00: A18 output (address) 01: A18 output (address) 10: Setting prohibited 11: A18 output (address) • On-chip ROM enabled extension mode 00: PB2 input/output (port) 01: TIA04 input (RSPI) 10: Setting prohibited 11: SSLA6 input (RSPI) • Single-chip mode 00: PB2 input/output (port) (initial value) 01: TIA04 input (RSPI) 10: Setting prohibited 11: SSLA6 input (RSPI) Note: This setting is valid only when PB2MD[1:0] = 00.
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
5, 4	PB1AMD [1:0]	00	R/W	PB1 Mode Control the function of the PB1/A17/FRRxD_B pin. • On-chip ROM disabled extension mode 00: A17 output (address) 01: Setting prohibited 10: Setting prohibited 11: A17 output (address) • On-chip ROM enabled extension mode 00: PB1 input/output (port) 01: Setting prohibited 10: Setting prohibited 11: FRRxD_B input (FlexRay) • Single-chip mode 00: PB1 input/output (port) (initial value) 01: Setting prohibited 10: Setting prohibited 11: FRRxD_B input (FlexRay) Note: This setting is valid only when PB1MD[1:0] = 00.
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
1, 0	PB0AMD [1:0]	00	R/W	PB0 Mode Control the function of the PB0/A16/FRTxD_B pin. • On-chip ROM disabled extension mode 00: A16 output (address) 01: Setting prohibited 10: Setting prohibited 11: A16 output (address) • On-chip ROM enabled extension mode 00: PB0 input/output (port) 01: Setting prohibited 10: Setting prohibited 11: FRTxD_B output (FlexRay) • Single-chip mode 00: PB0 input/output (port) (initial value) 01: Setting prohibited 10: Setting prohibited 11: FRTxD_B output (FlexRay) Note: This setting is valid only when PB0MD[1:0] = 00.

23.1.7 Port C I/O Register (PCIOR)

PCIOR is a 16-bit readable/writable register that sets the I/O direction of the port C pins.

PCIOR is enabled only when the port C pins function as general I/O pins. Otherwise, the set values of this register have no effect on the pin status.

PCIOR is initialized to H'0000 either by power-on reset, hardware standby, or power-on reset by the WDT. However, it is not initialized in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PC15 IOR	PC14 IOR	PC13 IOR	PC12 IOR	PC11 IOR	PC10 IOR	PC9 IOR	PC8 IOR	PC7 IOR	PC6 IOR	PC5 IOR	PC4 IOR	PC3 IOR	PC2 IOR	PC1 IOR	PC0 IOR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	PC15IOR to PC0IOR	All 0	R/W	The PC15IOR to PC0IOR bits correspond to the PC15 to PC0 pins respectively (the multiplex pin names other than the port names are omitted from pin names). 0: The corresponding pin is set to input. 1: The corresponding pin is set to output.

23.1.8 Port C Control Registers 1 to 4 (PCCR1 to PCCR4)

PCCR1 to PCCR4 are 16-bit readable/writable registers that control the functions of multiplexed pins of port C.

PCCR1 to PCCR4 are initialized to the values shown in table 23.15 either by power-on reset, hardware standby, or power-on reset by the WDT. However, they are not initialized in sleep mode.

Table 23.15 Initial Values of Port C Control Registers

	Initial Values			
	On-Chip ROM Disabled Extension Mode		On-Chip ROM Enabled Extension Mode	Single-Chip Mode
Register Name	**Area 0: 8 Bits**	**Area 0: 16 Bits**		
PCCR4	H'0000	H'1111	H'1111	H'0000
PCCR3	H'0000	H'1111	H'1111	H'0000
PCCR2	H'1111	H'1111	H'1111	H'0000
PCCR1	H'1111	H'1111	H'1111	H'0000

(1) Port C Control Register 4 (PCCR4)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	PC15 MD	-	-	-	PC14 MD	-	-	-	PC13 MD	-	-	-	PC12 MD
Initial value:	0	0	0	0/1*[1]	0	0	0	0/1*[1]	0	0	0	0/1*[1]	0	0	0	0/1*[1]
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Note: 1. The initial value depends on the operating mode of the LSI.

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	PC15MD	0/1*[1]	R/W	PC15 Mode Controls the function of the PC15/D15 pin. • On-chip ROM disabled extension mode (area 0: 8 bits) 0: PC15 input/output (port) (initial value) 1*[2]: D15 input/output (BSC) • On-chip ROM disabled extension mode (area 0: 16 bits) 0*[2]: D15 input/output (BSC) 1*[2]: D15 input/output (BSC) (initial value) • On-chip ROM enabled extension mode 0: PC15 input/output (port) 1*[2]: D15 input/output (BSC) (initial value) • Single-chip mode 0: PC15 input/output (port) (initial value) 1: PC15 input/output (port)
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
8	PC14MD	0/1*[1]	R/W	PC14 Mode Controls the function of the PC14/D14 pin. • On-chip ROM disabled extension mode (area 0: 8 bits) 0: PC14 input/output (port) (initial value) 1*[2]: D14 input/output (BSC) • On-chip ROM disabled extension mode (area 0: 16 bits) 0*[2]: D14 input/output (BSC) 1*[2]: D14 input/output (BSC) (initial value) • On-chip ROM enabled extension mode 0: PC14 input/output (port) 1*[2]: D14 input/output (BSC) (initial value) • Single-chip mode 0: PC14 input/output (port) (initial value) 1: PC14 input/output (port)
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
4	PC13MD	0/1*[1]	R/W	PC13 Mode Controls the function of the PC13/D13 pin. • On-chip ROM disabled extension mode (area 0: 8 bits) 0: PC13 input/output (port) (initial value) 1*[2]: D13 input/output (BSC) • On-chip ROM disabled extension mode (area 0: 16 bits) 0*[2]: D13 input/output (BSC) 1*[2]: D13 input/output (BSC) (initial value) • On-chip ROM enabled extension mode 0: PC13 input/output (port) 1*[2]: D13 input/output (BSC) (initial value) • Single-chip mode 0: PC13 input/output (port) (initial value) 1: PC13 input/output (port)
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
0	PC12MD	0/1*1	R/W	PC12 Mode Controls the function of the PC12/D12 pin. • On-chip ROM disabled extension mode (area 0: 8 bits) 0: PC12 input/output (port) (initial value) 1*2: D12 input/output (BSC) • On-chip ROM disabled extension mode (area 0: 16 bits) 0*2: D12 input/output (BSC) 1*2: D12 input/output (BSC) (initial value) • On-chip ROM enabled extension mode 0: PC12 input/output (port) 1*2: D12 input/output (BSC) (initial value) • Single-chip mode 0: PC12 input/output (port) (initial value) 1: PC12 input/output (port)

Notes: 1. The initial value depends on the operating mode of the LSI.

2. The pin is pulled up.

(2) Port C Control Register 3 (PCCR3)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	PC11 MD	-	-	-	PC10 MD	-	-	-	PC9 MD	-	-	-	PC8 MD
Initial value:	0	0	0	0/1*1	0	0	0	0/1*1	0	0	0	0/1*1	0	0	0	0/1*1
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Note: 1. The initial value depends on the operating mode of the LSI.

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	PC11MD	0/1*1	R/W	PC11 Mode Controls the function of the PC11/D11 pin. • On-chip ROM disabled extension mode (area 0: 8 bits) 0: PC11 input/output (port) (initial value) 1*2: D11 input/output (BSC) • On-chip ROM disabled extension mode (area 0: 16 bits) 0*2: D11 input/output (BSC) 1*2: D11 input/output (BSC) (initial value) • On-chip ROM enabled extension mode 0: PC11 input/output (port) 1*2: D11 input/output (BSC) (initial value) • Single-chip mode 0: PC11 input/output (port) (initial value) 1: PC11input/output (port)
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
8	PC10MD	0/1*[1]	R/W	PC10 Mode Controls the function of the PC10/D10 pin. • On-chip ROM disabled extension mode (area 0: 8 bits) 0: PC10 input/output (port) (initial value) 1*[2]: D10 input/output (BSC) • On-chip ROM disabled extension mode (area 0: 16 bits) 0*[2]: D10 input/output (BSC) 1*[2]: D10 input/output (BSC) (initial value) • On-chip ROM enabled extension mode 0: PC10 input/output (port) 1*[2]: D10 input/output (BSC) (initial value) • Single-chip mode 0: PC10 input/output (port) (initial value) 1: PC10 input/output (port)
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
4	PC9MD	0/1*[1]	R/W	PC9 Mode Controls the function of the PC9/D9 pin. • On-chip ROM disabled extension mode (area 0: 8 bits) 0: PC9 input/output (port) (initial value) 1*[2]: D9 input/output (BSC) • On-chip ROM disabled extension mode (area 0: 16 bits) 0*[2]: D9 input/output (BSC) 1*[2]: D9 input/output (BSC) (initial value) • On-chip ROM enabled extension mode 0: PC9 input/output (port) 1*[2]: D9 input/output (BSC) (initial value) • Single-chip mode 0: PC9 input/output (port) (initial value) 1: PC9 input/output (port)
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
0	PC8MD	0/1*[1]	R/W	PC8 Mode Controls the function of the PC8/D8 pin. • On-chip ROM disabled extension mode (area 0: 8 bits) 0: PC8 input/output (port) (initial value) 1*[2]: D8 input/output (BSC) • On-chip ROM disabled extension mode (area 0: 16 bits) 0*[2]: D8 input/output (BSC) 1*[2]: D8 input/output (BSC) (initial value) • On-chip ROM enabled extension mode 0: PC8 input/output (port) 1*[2]: D8 input/output (BSC) (initial value) • Single-chip mode 0: PC8 input/output (port) (initial value) 1: PC8 input/output (port)

Notes: 1. The initial value depends on the operating mode of the LSI.

2. The pin is pulled up.

(3) Port C Control Register 2 (PCCR2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	PC7 MD	-	-	-	PC6 MD	-	-	-	PC5 MD	-	-	-	PC4 MD
Initial value:	0	0	0	0/1*1	0	0	0	0/1*1	0	0	0	0/1*1	0	0	0	0/1*1
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Note: 1. The initial value depends on the operating mode of the LSI.

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	PC7MD	0/1*1	R/W	PC7 Mode Controls the function of the PC7/D7 pin. • On-chip ROM disabled extension mode 0*2: D7 input/output (BSC) 1*2: D7 input/output (BSC) (initial value) • On-chip ROM enabled extension mode 0: PC7 input/output (port) 1*2: D7 input/output (BSC) (initial value) • Single-chip mode 0: PC7 input/output (port) (initial value) 1: PC7 input/output (port)
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
8	PC6MD	0/1*[1]	R/W	PC6 Mode Controls the function of the PC6/D6 pin. • On-chip ROM disabled extension mode 0*[2]: D6 input/output (BSC) 1*[2]: D6 input/output (BSC) (initial value) • On-chip ROM enabled extension mode 0: PC6 input/output (port) 1*[2]: D6 input/output (BSC) (initial value) • Single-chip mode 0: PC6 input/output (port) (initial value) 1: PC6 input/output (port)
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	PC5MD	0/1*[1]	R/W	PC5 Mode Controls the function of the PC5/D5 pin. • On-chip ROM disabled extension mode 0*[2]: D5 input/output (BSC) 1*[2]: D5 input/output (BSC) (initial value) • On-chip ROM enabled extension mode 0: PC5 input/output (port) 1*[2]: D5 input/output (BSC) (initial value) • Single-chip mode 0: PC5 input/output (port) (initial value) 1: PC5 input/output (port)
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
0	PC4MD	0/1*[1]	R/W	PC4 Mode Controls the function of the PC4/D4 pin. • On-chip ROM disabled extension mode 0*[2]: D4 input/output (BSC) 1*[2]: D4 input/output (BSC) (initial value) • On-chip ROM enabled extension mode 0: PC4 input/output (port) 1*[2]: D4 input/output (BSC) (initial value) • Single-chip mode 0: PC4 input/output (port) (initial value) 1: PC4 input/output (port)

Notes: 1. The initial value depends on the operating mode of the LSI.

2. The pin is pulled up.

(4) Port C Control Register 1 (PCCR1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	PC3 MD	-	-	-	PC2 MD	-	-	-	PC1 MD	-	-	-	PC0 MD
Initial value:	0	0	0	0/1*[1]	0	0	0	0/1*[1]	0	0	0	0/1*[1]	0	0	0	0/1*[1]
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Note: 1. The initial value depends on the operating mode of the LSI.

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	PC3MD	0/1*[1]	R/W	PC3 Mode Controls the function of the PC3/D3 pin. • On-chip ROM disabled extension mode 0*[2]: D3 input/output (BSC) 1*[2]: D3 input/output (BSC) (initial value) • On-chip ROM enabled extension mode 0: PC3 input/output (port) 1*[2]: D3 input/output (BSC) (initial value) • Single-chip mode 0: PC3 input/output (port) (initial value) 1: PC3 input/output (port)
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
8	PC2MD	0/1*[1]	R/W	PC2 Mode Controls the function of the PC2/D2 pin. • On-chip ROM disabled extension mode 0*[2]: D2 input/output (BSC) 1*[2]: D2 input/output (BSC) (initial value) • On-chip ROM enabled extension mode 0: PC2 input/output (port) 1*[2]: D2 input/output (BSC) (initial value) • Single-chip mode 0: PC2 input/output (port) (initial value) 1: PC2 input/output (port)
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	PC1MD	0/1*[1]	R/W	PC1 Mode Controls the function of the PC1/D1 pin. • On-chip ROM disabled extension mode 0*[2]: D1 input/output (BSC) 1*[2]: D1 input/output (BSC) (initial value) • On-chip ROM enabled extension mode 0: PC1 input/output (port) 1*[2]: D1 input/output (BSC) (initial value) • Single-chip mode 0: PC1 input/output (port) (initial value) 1: PC1 input/output (port)
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
0	PC0MD	0/1*[1]	R/W	PC0 Mode Controls the function of the PC0/D0 pin. • On-chip ROM disabled extension mode 0*[2]: D0 input/output (BSC) 1*[2]: D0 input/output (BSC) (initial value) • On-chip ROM enabled extension mode 0: PC0 input/output (port) 1*[2]: D0 input/output (BSC) (initial value) • Single-chip mode 0: PC0 input/output (port) (initial value) 1: PC0 input/output (port)

Notes: 1. The initial value depends on the operating mode of the LSI.

2. The pin is pulled up.

23.1.9 Port C Control Registers 2A to 4A (PCCR2A to PCCR4A)

PCCR2A to PCCR4A are 16-bit readable/writable registers that control the functions of multiplexed pins of port C.

PCCR2A to PCCR4A are initialized to H'0000 either by power-on reset, hardware standby, or power-on reset by the WDT. However, they are not initialized in sleep mode.

(1) Port C Control Register 4A (PCCR4A)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PC15AMD[1:0]		-	-	PC14AMD[1:0]		-	-	PC13AMD[1:0]		-	-	PC12AMD[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
13, 12	PC15AMD [1:0]	00	R/W	PC15 Mode Controls the function of the PC15/D15/TOD13A/SSLA3 pin. • On-chip ROM disabled extension mode (area 0: 8 bits) 00: PC15 input/output (port) (initial value) 01: TOD13A output (ATU-III) 10: Setting prohibited 11: SSLA3 output (RSPI) • On-chip ROM disabled extension mode (area 0: 16 bits) 00*: D15 input/output (BSC) 01*: D15 input/output (BSC) 10: Setting prohibited 11*: D15 input/output (BSC) • On-chip ROM enabled extension mode 00: PC15 input/output (port) 01: TOD13A output (ATU-III) 10: Setting prohibited 11: SSLA3 output (RSPI) • Single-chip mode 00: PC15 input/output (port) (initial value) 01: TOD13A output (ATU-III) 10: Setting prohibited 11: SSLA3 output (RSPI) Note: This setting is valid only when PC15MD = 0.
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
9, 8	PC14AMD [1:0]	00	R/W	PC14 Mode Controls the function of the PC14/D14/TOD12A/SSLA2 pin. • On-chip ROM disabled extension mode (area 0: 8 bits) 00: PC14 input/output (port) (initial value) 01: TOD12A output (ATU-III) 10: Setting prohibited 11: SSLA2 output (RSPI) • On-chip ROM disabled extension mode (area 0: 16 bits) 00*: D14 input/output (BSC) 01*: D14 input/output (BSC) 10: Setting prohibited 11*: D14 input/output (BSC) • On-chip ROM enabled extension mode 00: PC14 input/output (port) 01: TOD12A output (ATU-III) 10: Setting prohibited 11: SSLA2 output (RSPI) • Single-chip mode 00: PC14 input/output (port) (initial value) 01: TOD12A output (ATU-III) 10: Setting prohibited 11: SSLA2 output (RSPI) Note: This setting is valid only when PC14MD = 0.
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
5, 4	PC13AMD [1:0]	00	R/W	PC13 Mode Controls the function of the PC13/D13/TOD11A/SSLA1 pin. • On-chip ROM disabled extension mode (area 0: 8 bits) 00: PC13 input/output (port) (initial value) 01: TOD11A output (ATU-III) 10: Setting prohibited 11: SSLA1 output (RSPI) • On-chip ROM disabled extension mode (area 0: 16 bits) 00*: D13 input/output (BSC) 01*: D13 input/output (BSC) 10: Setting prohibited 11*: D13 input/output (BSC) • On-chip ROM enabled extension mode 00: PC13 input/output (port) 01: TOD11A output (ATU-III) 10: Setting prohibited 11: SSLA1 output (RSPI) • Single-chip mode 00: PC13 input/output (port) (initial value) 01: TOD11A output (ATU-III) 10: Setting prohibited 11: SSLA1 output (RSPI) Note: This setting is valid only when PC13MD = 0.
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
1, 0	PC12AMD [1:0]	00	R/W	PC12 Mode Controls the function of the PC12/D12/TOD10A/SSLA0 pin. • On-chip ROM disabled extension mode (area 0: 8 bits) 00: PC12 input/output (port) (initial value) 01: TOD10A output (ATU-III) 10: Setting prohibited 11: SSLA0 input/output (RSPI) • On-chip ROM disabled extension mode (area 0: 16 bits) 00*: D12 input/output (BSC) 01*: D12 input/output (BSC) 10: Setting prohibited 11*: D12 input/output (BSC) • On-chip ROM enabled extension mode 00: PC12 input/output (port) 01: TOD10A output (ATU-III) 10: Setting prohibited 11: SSLA0 input/output (RSPI) • Single-chip mode 00: PC12 input/output (port) (initial value) 01: TOD10A output (ATU-III) 10: Setting prohibited 11: SSLA0 input/output (RSPI) Note: This setting is valid only when PC12MD = 0.

Notes: * The pin is pulled up.

(2) Port C Control Register 3A (PCCR3A)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PC11AMD[1:0]		-	-	PC10AMD[1:0]		-	-	PC9AMD[1:0]		-	-	PC8AMD[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
13, 12	PC11AMD [1:0]	00	R/W	PC11 Mode Controls the function of the PC11/D11/TOD03A/SSLB3 pin. • On-chip ROM disabled extension mode (area 0: 8 bits) 00: PC11 input/output (port) (initial value) 01: TOD03A output (ATU-III) 10: Setting prohibited 11: SSLB3 output (RSPI) • On-chip ROM disabled extension mode (area 0: 16 bits) 00*: D11 input/output (BSC) 01*: D11 input/output (BSC) 10: Setting prohibited 11*: D11 input/output (BSC) • On-chip ROM enabled extension mode 00: PC11 input/output (port) 01: TOD03A output (ATU-III) 10: Setting prohibited 11: SSLB3 output (RSPI) • Single-chip mode 00: PC11 input/output (port) (initial value) 01: TOD03A output (ATU-III) 10: Setting prohibited 11: SSLB3 output (RSPI) Note: This setting is valid only when PC11MD = 0.
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
9, 8	PC10AMD [1:0]	00	R/W	PC10 Mode Controls the function of the PC10/D10/TOD02A/SCK_A pin. • On-chip ROM disabled extension mode (area 0: 8 bits) 00: PC10 input/output (port) (initial value) 01: TOD02A output (ATU-III) 10: SCK_A input/output (SCI) 11: Setting prohibited • On-chip ROM disabled extension mode (area 0: 16 bits) 00*: D10 input/output (BSC) 01*: D10 input/output (BSC) 10*: D10 input/output (BSC) 11: Setting prohibited • On-chip ROM enabled extension mode 00: PC10 input/output (port) 01: TOD02A output (ATU-III) 10: SCK_A input/output (SCI) 11: Setting prohibited • Single-chip mode 00: PC10 input/output (port) (initial value) 01: TOD02A output (ATU-III) 10: SCK_A input/output (SCI) 11: Setting prohibited Note: This setting is valid only when PC10MD = 0.
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
5, 4	PC9AMD [1:0]	00	R/W	PC9 Mode Controls the function of the PC9/D/9/TOD01A/RxD_A/CRx_A pin. • On-chip ROM disabled extension mode (area 0: 8 bits) 00: PC9 input/output (port) (initial value) 01: TOD01A output (ATU-III) 10: RxD_A input (SCI) 11: CRx_A input (RCAN) • On-chip ROM disabled extension mode (area 0: 16 bits) 00*: D9 input/output (BSC) 01*: D9 input/output (BSC) 10*: D9 input/output (BSC) 11*: CRx_A input (RCAN) • On-chip ROM enabled extension mode 00: PC9 input/output (port) 01: TOD01A output (ATU-III) 10: RxD_A input (SCI) 11: CRx_A input (RCAN) • Single-chip mode 00: PC9 input/output (port) (initial value) 01: TOD01A output (ATU-III) 10: RxD_A input (SCI) 11: CRx_A input (RCAN) Note: This setting is valid only when PC9MD = 0.
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
1, 0	PC8AMD [1:0]	00	R/W	PC8 Mode Controls the function of the PC8/D8/TOD00A/TxD_A/CTx_A pin. • On-chip ROM disabled extension mode (area 0: 8 bits) 00: PC8 input/output (port) (initial value) 01: TOD00A output (ATU-III) 10: TxD_A output (SCI) 11: CTx_A output (RCAN) • On-chip ROM disabled extension mode (area 0: 16 bits) 00*: D8 input/output (BSC) 01*: D8 input/output (BSC) 10*: D8 input/output (BSC) 11*: D8 input/output (BSC) • On-chip ROM enabled extension mode 00: PC8 input/output (port) 01: TOD00A output (ATU-III) 10: TxD_A output (SCI) 11: CTx_A output (RCAN) • Single-chip mode 00: PC8 input/output (port) (initial value) 01: TOD00A output (ATU-III) 10: TxD_A output (SCI) 11: CTx_A output (RCAN) Note: This setting is valid only when PC8MD = 0.

Notes: * The pin is pulled up.

(3) Port C Control Register 2A (PCCR2A)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PC7AMD[1:0]		-	-	PC6AMD[1:0]		-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13, 12	PC7AMD [1:0]	00	R/W	PC7 Mode Controls the function of the PC7/D7/ADEND_A pin. • On-chip ROM disabled extension mode 00*: D7 input/output (BSC) 01: Setting prohibited 10: Setting prohibited 11*: D7 input/output (BSC) • On-chip ROM enabled extension mode 00: PC7 input/output (port) 01: Setting prohibited 10: Setting prohibited 11: ADEND_A output (ADC) • Single-chip mode 00: PC7 input/output (port) (initial value) 01: Setting prohibited 10: Setting prohibited 11: ADEND_A output (ADC) Note: This setting is valid only when PC7MD = 0.
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
9, 8	PC6AMD [1:0]	00	R/W	PC6 Mode Controls the function of the PC6/D6/ADTRG_A pin. • On-chip ROM disabled extension mode 00*: D6 input/output (BSC) 01: Setting prohibited 10: Setting prohibited 11*: D6 input/output (BSC) • On-chip ROM enabled extension mode 00: PC6 input/output (port) 01: Setting prohibited 10: Setting prohibited 11: ADTRG_A input (ADC) • Single-chip mode 00: PC6 input/output (port) (initial value) 01: Setting prohibited 10: Setting prohibited 11: ADTRG_A input (ADC) Note: This setting is valid only when PC6MD = 0.
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Notes: * The pin is pulled up.

23.1.10 Port D I/O Register (PDIOR)

PDIOR is a 16-bit readable/writable register that sets the I/O direction of the port D pins.

PDIOR is enabled only when the port D pins function as general I/O pins. Otherwise, the set values of this register have no effect on the pin status.

PDIOR is initialized to H'0000 either by power-on reset, hardware standby, or power-on reset by the WDT. However, it is not initialized in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PD13 IOR	PD12 IOR	PD11 IOR	PD10 IOR	PD9 IOR	PD8 IOR	PD7 IOR	PD6 IOR	PD5 IOR	PD4 IOR	PD3 IOR	PD2 IOR	PD1 IOR	PD0 IOR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13 to 0	PD13IOR to PD0IOR	All 0	R/W	The PD13IOR to PD0IOR bits correspond to the PD13 to PC0 pins respectively (the multiplex pin names other than the port names are omitted from pin names). 0: The corresponding pin is set to input. 1: The corresponding pin is set to output.

23.1.11 Port D Control Registers 1 and 2 (PDCR1 and PDCR2)

PDCR1 and PDCR2 are 16-bit readable/writable registers that control the functions of multiplexed pins of port D.

PDCR1 and PDCR2 are initialized to H'0000 either by power-on reset, hardware standby, or power-on reset by the WDT. However, they are not initialized in sleep mode.

(1) Port D Control Register 2 (PDCR2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	PD13MD[1:0]		PD12MD[1:0]		PD11MD[1:0]		PD10MD[1:0]		PD9MD[1:0]		PD8MD[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11, 10	PD13MD [1:0]	00	R/W	PD13 Mode Control the function of the PD13/TCLKB/TIJ1 pin. 00: PD13 input/output (port) 01: TCLKB input (ATU-III) 10: Setting prohibited 11: TIJ1 input (ATU-III)
9, 8	PD12MD [1:0]	00	R/W	PD12 Mode Control the function of the PD12/TCLKA/TIOC41/TIJ0 pin. 00: PD12 input/output (port) 01: TCLKA input (ATU-III) 10: TIOC41 input/output (ATU-III) 11: TIJ0 input (ATU-III)

Bit	Bit Name	Initial Value	R/W	Description
7, 6	PD11MD [1:0]	00	R/W	PD11 Mode Control the function of the PD11/TIOC23/TIF2B/TOE51 pin. 00: PD11 input/output (port) 01: TIOC23 input/output (ATU-III) 10: TIF2B input (ATU-III) 11: TOE51 output (ATU-III)
5, 4	PD10MD [1:0]	00	R/W	PD10 Mode Control the function of the PD10/TIOC22/TIF1B/TOE50 pin. 00: PD10 input/output (port) 01: TIOC22 input/output (ATU-III) 10: TIF1B input (ATU-III) 11: TOE50 output (ATU-III)
3, 2	PD9MD [1:0]	00	R/W	PD9 Mode Control the function of the PD9/TIOC21/TIF0B/TOE43 pin. 00: PD9 input/output (port) 01: TIOC21 input/output (ATU-III) 10: TIF0B input (ATU-III) 11: TOE43 output (ATU-III)
1, 0	PD8MD [1:0]	00	R/W	PD8 Mode Control the function of the PD8/TIOC20/TIOC33/TOE53 pin. 00: PD8 input/output (port) 01: TIOC20 input/output (ATU-III) 10: TIOC33 input/output (ATU-III) 11: TOE53 output (ATU-III)

(2) Port D Control Register 1 (PDCR1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PD7MD[1:0]		PD6MD[1:0]		PD5MD[1:0]		PD4MD[1:0]		PD3MD[1:0]		PD2MD[1:0]		PD1MD[1:0]		PD0MD[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	PD7MD [1:0]	00	R/W	PD7 Mode Control the function of the PD7/TIOC13/TOE42 pin. 00: PD7 input/output (port) 01: TIOC13 input/output (ATU-III) 10: Setting prohibited 11: TOE42 output (ATU-III)
13, 12	PD6MD [1:0]	00	R/W	PD6 Mode Control the function of the PD6/TIOC12/TOE41 pin. 00: PD6 input/output (port) 01: TIOC12 input/output (ATU-III) 10: Setting prohibited 11: TOE41 output (ATU-III)
11, 10	PD5MD [1:0]	00	R/W	PD5 Mode Control the function of the PD5/TIOC11/TOE23/TOE40 pin. 00: PD5 input/output (port) 01: TIOC11 input/output (ATU-III) 10: TOE23 output (ATU-III) 11: TOE40 output (ATU-III)
9, 8	PD4MD [1:0]	00	R/W	PD4 Mode Control the function of the PD4/TIOC10/TIOC32/TOE52 pin. 00: PD4 input/output (port) 01: TIOC10 input/output (ATU-III) 10: TIOC32 input/output (ATU-III) 11: TOE52 output (ATU-III)

Bit	Bit Name	Initial Value	R/W	Description
7, 6	PD3MD [1:0]	00	R/W	PD3 Mode Control the function of the PD3/TIOC03/TOE22/TOE53 pin. 00: PD3 input/output (port) 01: TIOC03 input/output (ATU-III) 10: TOE22 output (ATU-III) 11: TOE53 output (ATU-III)
5, 4	PD2MD [1:0]	00	R/W	PD2 Mode Control the function of the PD2/TIOC02/TOE21/TOE52 pin. 00: PD2 input/output (port) 01: TIOC02 input/output (ATU-III) 10: TOE21 output (ATU-III) 11: TOE52 output (ATU-III)
3, 2	PD1MD [1:0]	00	R/W	PD1 Mode Control the function of the PD1/TIOC01/TOE20 pin. 00: PD1 input/output (port) 01: TIOC01 input/output (ATU-III) 10: TOE20 output (ATU-III) 11: Setting prohibited
1, 0	PD0MD [1:0]	00	R/W	PD0 Mode Control the function of the PD0/TIOC00/TIOC31 pin. 00: PD0 input/output (port) 01: TIOC00 input/output (ATU-III) 10: TIOC31 input/output (ATU-III) 11: Setting prohibited

23.1.12 Port D Control Registers 2A (PDCR2A)

PDCR2A is 16-bit readable/writable register that controls the functions of multiplexed pins of port D.

PDCR2A is initialized to H'0000 either by power-on reset, hardware standby, or power-on reset by the WDT. However, it is not initialized in sleep mode.

(1) Port D Control Register 2A (PDCR2A)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	PD13A MD	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10	PD13AMD	0	R/W	PD13 Mode Control the function of the PD13/TOE60 pin. 0: PD13 input/output (port) 1: TOE60 output (ATU-III) Note: This setting is valid only when PD13MD[1:0] = 0.
9 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

23.1.13 Port E I/O Register (PEIOR)

PEIOR is a 16-bit readable/writable register that sets the I/O direction of the port E pins.

PEIOR is enabled only when the port E pins function as general I/O pins. Otherwise, the set values of this register have no effect on the pin status.

PEIOR is initialized to H'0000 either by power-on reset, hardware standby, or power-on reset by the WDT. However, it is not initialized in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PE13 IOR	PE12 IOR	PE11 IOR	PE10 IOR	PE9 IOR	PE8 IOR	PE7 IOR	PE6 IOR	PE5 IOR	PE4 IOR	PE3 IOR	PE2 IOR	PE1 IOR	PE0 IOR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13 to 0	PE13IOR to PE0IOR	All 0	R/W	The PE13IOR to PE0IOR bits correspond to the PE13 to PE0 pins respectively (the multiplex pin names other than the port names are omitted from pin names). 0: The corresponding pin is set to input. 1: The corresponding pin is set to output.

23.1.14 Port E Control Registers 1 and 2 (PECR1 and PECR2)

PECR1 and PECR2 are 16-bit readable/writable registers that control the functions of multiplexed pins of port E.

PECR1 and PECR2 are initialized to H'0000 either by power-on reset, hardware standby, or power-on reset by the WDT. However, they are not initialized in sleep mode.

(1) Port E Control Register 2 (PECR2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	PE13 MD	-	PE12 MD	-	PE11 MD	-	PE10 MD	-	PE9 MD	-	PE8 MD
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R	R/W	R	R/W	R	R/W	R	R/W	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10	PE13MD	0	R/W	PE13 Mode Controls the function of the PE13/TOE13 pin. 0: PE13 input/output (port) 1: TOE13 output (ATU-III)
9	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
8	PE12MD	0	R/W	PE12 Mode Controls the function of the PE12/TOE12 pin. 0: PE12 input/output (port) 1: TOE12 output (ATU-III)
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
6	PE11MD	0	R/W	PE11 Mode Controls the function of the PE11/TOE11 pin. 0: PE11 input/output (port) 1: TOE11 output (ATU-III)
5	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
4	PE10MD	0	R/W	PE10 Mode Controls the function of the PE10/TOE10 pin. 0: PE10 input/output (port) 1: TOE10 output (ATU-III)
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	PE9MD	0	R/W	PE9 Mode Controls the function of the PE9/TOE03 pin. 0: PE9 input/output (port) 1: TOE03 output (ATU-III)
1	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
0	PE8MD	0	R/W	PE8 Mode Controls the function of the PE8/TOE02 pin. 0: PE8 input/output (port) 1: TOE02 output (ATU-III)

(2) Port E Control Register 1 (PECR1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PE7MD[1:0]		PE6MD[1:0]		-	PE5 MD	-	PE4 MD	-	PE3 MD	PE2MD[1:0]		PE1MD[1:0]		-	PE0 MD
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R/W	R	R/W	R	R/W	R/W	R/W	R/W	R/W	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	PE7MD [1:0]	00	R/W	PE7 Mode Control the function of the PE7/TOE01/CRx_B pin. 00: PE7 input/output (port) 01: TOE01 output (ATU-III) 10: CRx_B input (RCAN-TL1) 11: Setting prohibited
13, 12	PE6MD [1:0]	00	R/W	PE6 Mode Control the function of the PE6/TOE00/CTx_B pin. 00: PE6 input/output (port) 01: TOE00 output (ATU-III) 10: CTx_B output (RCAN-TL1) 11: Setting prohibited
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10	PE5MD	0	R/W	PE5 Mode Controls the function of the PE5/TIA05 pin. 0: PE5 input/output (port) 1: TIA05 input (ATU-III)
9	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
8	PE4MD	0	R/W	PE4 Mode Controls the function of the PE4/TIA04 pin. 0: PE4 input/output (port) 1: TIA04 input (ATU-III)

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6	PE3MD	0	R/W	PE3 Mode Controls the function of the PE3/TIA03 pin. 0: PE3 input/output (port) 1: TIA03 input (ATU-III)
5, 4	PE2MD [1:0]	00	R/W	PE2 Mode Control the function of the PE2/TIA02/TIOC43/TIOC30 pin. 00: PE2 input/output (port) 01: TIA02 input (ATU-III) 10: TIOC43 input/output (ATU-III) 11: TIOC30 input/output (ATU-III)
3, 2	PE1MD [1:0]	00	R/W	PE1 Mode Control the function of the PE1/TIA01/TIOC42/TIOC40 pin. 00: PE1 input/output (port) 01: TIA01 input (ATU-III) 10: TIOC42 input/output (ATU-III) 11: TIOC40 input/output (ATU-III)
1	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
0	PE0MD	0	R/W	PE0 Mode Controls the function of the PE0/TIA00 pin. 0: PE0 input/output (port) 1: TIA00 input (ATU-III)

23.1.15 Port E Control Registers 1A (PECR1A)

PECR1A is 16-bit readable/writable register that controls the functions of multiplexed pins of port E.

PECR1A is initialized to H'0000 either by power-on reset, hardware standby, or power-on reset by the WDT. However, it is not initialized in sleep mode.

(1) Port E Control Register 1A (PECR1A)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	PE5AMD[1:0]		PE4AMD[1:0]		PE3AMD[1:0]		-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11, 10	PE5AMD [1:0]	00	R/W	PE5 Mode Controls the function of the PE5/TOE63/TIF25 pin. 00: PE5 input/output (port) 01: TOE63 output (ATU-III) 10: TIF25 input (ATU-III) 11: Setting prohibited Note: This setting is valid only when PE5MD = 0.
9, 8	PE4AMD [1:0]	00	R/W	PE4 Mode Controls the function of the PE4/TOE62/TIF24 pin. 00: PE4 input/output (port) 01: TOE62 output (ATU-III) 10: TIF24 input (ATU-III) 11: Setting prohibited Note: This setting is valid only when PE4MD = 0.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	PE3AMD [1:0]	00	R/W	PE3 Mode Controls the function of the PE3/ TOE61/TIF23 pin. 00: PE3 input/output (port) 01: TOE61 output (ATU-III) 10: TIF23 input (ATU-III) 11: Setting prohibited Note: This setting is valid only when PE3MD = 0.
5 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

23.1.16 Port F I/O Register (PFIOR)

PFIOR is a 16-bit readable/writable register that sets the I/O direction of the port F pins.

PFIOR is enabled only when the port F pins function as general I/O pins. Otherwise, the set values of this register have no effect on the pin status.

PFIOR is initialized to H'0000 either by power-on reset, hardware standby, or power-on reset by the WDT. However, it is not initialized in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PF15 IOR	PF14 IOR	PF13 IOR	PF12 IOR	PF11 IOR	PF10 IOR	PF9 IOR	PF8 IOR	PF7 IOR	PF6 IOR	PF5 IOR	PF4 IOR	PF3 IOR	PF2 IOR	PF1 IOR	PF0 IOR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	PF15IOR to PF0IOR	All 0	R/W	The PF15IOR to PF0IOR bits correspond to the PF15 to PF0 pins respectively (the multiplex pin names other than the port names are omitted from pin names). 0: The corresponding pin is set to input. 1: The corresponding pin is set to output.

23.1.17 Port F Control Registers 1 and 2 (PFCR1 and PFCR2)

PFCR1 and PFCR2 are 16-bit readable/writable registers that control the functions of multiplexed pins of port F.

PFCR1 and PFCR2 are initialized to H'0000 either by power-on reset, hardware standby, or power-on reset by the WDT. However, they are not initialized in sleep mode.

(1) Port F Control Register 2 (PFCR2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PF15MD[1:0]		PF14MD[1:0]		PF13MD[1:0]		PF12MD[1:0]		PF11MD[1:0]		PF10MD[1:0]		PF9MD[1:0]		PF8MD[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	PF15MD [1:0]	00	R/W	PF15 Mode Control the function of the PF15/TOD33B/CRx_B/RxD_A pin. 00: PF15 input/output (port) 01: TOD33B output (ATU-III) 10: CRx_B input (RCAN-TL1) 11: RxD_A input (SCI)
13, 12	PF14MD [1:0]	00	R/W	PF14 Mode Control the function of the PF14/TOD32B/CTx_B/TxD_A pin. 00: PF14 input/output (port) 01: TOD32B output (ATU-III) 10: CTx_B output (RCAN-TL1) 11: TxD_A output (SCI)
11, 10	PF13MD [1:0]	00	R/W	PF13 Mode Control the function of the PF13/TOD31B/TIF19 pin. 00: PF13 input/output (port) 01: TOD31B output (ATU-III) 10: Setting prohibited 11: TIF19 input (ATU-III)

Bit	Bit Name	Initial Value	R/W	Description
9, 8	PF12MD [1:0]	00	R/W	PF12 Mode Control the function of the PF12/TOD30B/TIF18 pin. 00: PF12 input/output (port) 01: TOD30B output (ATU-III) 10: Setting prohibited 11: TIF18 input (ATU-III)
7, 6	PF11MD [1:0]	00	R/W	PF11 Mode Control the function of the PF11/TOD23B/TIF17 pin 00: PF11 input/output (port) 01: TOD23B output (ATU-III) 10: Setting prohibited 11: TIF17 input (ATU-III)
5, 4	PF10MD [1:0]	00	R/W	PF10 Mode Control the function of the PF10/TOD22B/TIF16 pin. 00: PF10 input/output (port) 01: TOD22B output (ATU-III) 10: Setting prohibited 11: TIF16 input (ATU-III)
3, 2	PF9MD [1:0]	00	R/W	PF9 Mode Control the function of the PF9/TOD21B/TIF15 pin. 00: PF9 input/output (port) 01: TOD21B output (ATU-III) 10: Setting prohibited 11: TIF15 input (ATU-III)
1, 0	PF8MD [1:0]	00	R/W	PF8 Mode Control the function of the PF8/TOD20B/TIF14 pin. 00: PF8 input/output (port) 01: TOD20B output (ATU-III) 10: Setting prohibited 11: TIF14 input (ATU-III)

(2) Port F Control Register 1 (PFCR1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PF7MD[1:0]		PF6MD[1:0]		PF5MD[1:0]		PF4MD[1:0]		PF3MD[1:0]		PF2MD[1:0]		PF1MD[1:0]		PF0MD[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	PF7MD [1:0]	00	R/W	PF7 Mode Control the function of the PF7/TOD13B/TIF13 pin. 00: PF7 input/output (port) 01: TOD13B output (ATU-III) 10: Setting prohibited 11: TIF13 input (ATU-III)
13, 12	PF6MD [1:0]	00	R/W	PF6 Mode Control the function of the PF6/TOD12B/TIF12 pin. 00: PF6 input/output (port) 01: TOD12B output (ATU-III) 10: Setting prohibited 11: TIF12 input (ATU-III)
11, 10	PF5MD [1:0]	00	R/W	PF5 Mode Control the function of the PF5/TOD11B/TIF11 pin. 00: PF5 input/output (port) 01: TOD11B output (ATU-III) 10: Setting prohibited 11: TIF11 input (ATU-III)
9, 8	PF4MD [1:0]	00	R/W	PF4 Mode Control the function of the PF4/TOD10B/TIF10 pin. 00: PF4 input/output (port) 01: TOD10B output (ATU-III) 10: Setting prohibited 11: TIF10 input (ATU-III)

Bit	Bit Name	Initial Value	R/W	Description
7, 6	PF3MD [1:0]	00	R/W	PF3 Mode Control the function of the PF3/TOD03B/TIF9 pin. 00: PF3 input/output (port) 01: TOD03B output (ATU-III) 10: Setting prohibited 11: TIF9 input (ATU-III)
5, 4	PF2MD [1:0]	00	R/W	PF2 Mode Control the function of the PF2/TOD02B/TIF8 pin. 00: PF2 input/output (port) 01: TOD02B output (ATU-III) 10: Setting prohibited 11: TIF8 input (ATU-III)
3, 2	PF1MD [1:0]	00	R/W	PF1 Mode Control the function of the PF1/TOD01B/TIF7 pin. 00: PF1 input/output (port) 01: TOD01B output (ATU-III) 10: Setting prohibited 11: TIF7 input (ATU-III)
1, 0	PF0MD [1:0]	00	R/W	PF0 Mode Control the function of the PF0/TOD00B/TIF6 pin. 00: PF0 input/output (port) 01: TOD00B output (ATU-III) 10: Setting prohibited 11: TIF6 input (ATU-III)

23.1.18 Port F Control Registers 2A (PFCR2A)

PFCR2A is 16-bit readable/writable register that controls the functions of multiplexed pins of port F.

PFCR2A is initialized to H'0000 either by power-on reset, hardware standby, or power-on reset by the WDT. However, it is not initialized in sleep mode.

(1) Port F Control Register 2A (PFCR2A)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PF15AMD[1:0]		PF14AMD[1:0]		-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15, 14	PF15AMD [1:0]	00	R/W	PF15 Mode Control the function of the PF15/FRRxD_B pin. 00: PF15 input/output (port) 01: Setting prohibited 10: Setting prohibited 11: FRRxD_B input (FlexRay) Note: This setting is valid only when PF15MD[1:0] = 00.
13, 12	PF14AMD [1:0]	00	R/W	PF14 Mode Control the function of the PF14/FRTxD_B pin. 00: PF14 input/output (port) 01: Setting prohibited 10: Setting prohibited 11: FRTxD_B output (FlexRay) Note: This setting is valid only when PF14MD[1:0] = 00.
11 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

23.1.19 Port G I/O Register (PGIOR)

PGIOR is a 16-bit readable/writable register that sets the I/O direction of the port G pins.

PGIOR is enabled only when the port G pins function as general I/O pins. Otherwise, the set values of this register have no effect on the pin status.

PGIOR is initialized to H'0000 either by power-on reset, hardware standby, or power-on reset by the WDT. However, it is not initialized in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PG15 IOR	PG14 IOR	PG13 IOR	PG12 IOR	PG11 IOR	PG10 IOR	PG9 IOR	PG8 IOR	PG7 IOR	PG6 IOR	PG5 IOR	PG4 IOR	PG3 IOR	PG2 IOR	PG1 IOR	PG0 IOR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	PG15IOR to PG0IOR	All 0	R/W	The PG15IOR to PG0IOR bits correspond to the PG15 to PG0 pins respectively (the multiplex pin names other than the port names are omitted from pin names). 0: The corresponding pin is set to input. 1: The corresponding pin is set to output.

23.1.20 Port G Control Registers 1 and 2 (PGCR1 and PGCR2)

PGCR1 and PGCR2 are 16-bit readable/writable registers that control the functions of multiplexed pins of port G.

PGCR1 and PGCR2 are initialized to H'0000 either by power-on reset, hardware standby, or power-on reset by the WDT. However, they are not initialized in sleep mode.

(1) Port G Control Register 2 (PGCR2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PG15MD[1:0]		PG14MD[1:0]		PG13MD[1:0]		PG12MD[1:0]		PG11MD[1:0]		PG10MD[1:0]		PG9MD[1:0]		PG8MD[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	PG15MD [1:0]	00	R/W	PG15 Mode Control the function of the PG15/TOD33A/SSLA7/TIF13 pin. 00: PG15 input/output (port) 01: TOD33A output (ATU-III) 10: SSLA7 output (RSPI) 11: TIF13 input (ATU-III)
13, 12	PG14MD [1:0]	00	R/W	PG14 Mode Control the function of the PG14/TOD32A/SSLA6/TIF12 pin. 00: PG14 input/output (port) 01: TOD32A output (ATU-III) 10: SSLA6 output (RSPI) 11: TIF12 input (ATU-III)

Bit	Bit Name	Initial Value	R/W	Description
11, 10	PG13MD [1:0]	00	R/W	PG13 Mode Control the function of the PG13/TOD31A/SSLA5/TIF11 pin. 00: PG13 input/output (port) 01: TOD31A output (ATU-III) 10: SSLA5 output (RSPI) 11: TIF11 input (ATU-III)
9, 8	PG12MD [1:0]	00	R/W	PG12 Mode Control the function of the PG12/TOD30A/SSLA4/TIF10 pin. 00: PG12 input/output (port) 01: TOD30A output (ATU-III) 10: SSLA4 output (RSPI) 11: TIF10 input (ATU-III)
7, 6	PG11MD [1:0]	00	R/W	PG11 Mode Control the function of the PG11/TOD23A/SSLC2/TIF9 pin. 00: PG11 input/output (port) 01: TOD23A output (ATU-III) 10: SSLC2 output (RSPI) 11: TIF9 input (ATU-III)
5, 4	PG10MD [1:0]	00	R/W	PG10 Mode Control the function of the PG10/TOD22A/SSLC1/TIF8 pin. 00: PG10 input/output (port) 01: TOD22A output (ATU-III) 10: SSLC1 output (RSPI) 11: TIF8 input (ATU-III)

Bit	Bit Name	Initial Value	R/W	Description
3, 2	PG9MD [1:0]	00	R/W	PG9 Mode Control the function of the PG9/TOD21A/SSLC0/TIF7 pin. 00: PG9 input/output (port) 01: TOD21A output (ATU-III) 10: SSLC0 input/output (RSPI) 11: TIF7 input (ATU-III)
1, 0	PG8MD [1:0]	00	R/W	PG8 Mode Control the function of the PG8/TOD20A/SSLB2/TIF6 pin. 00: PG8 input/output (port) 01: TOD20A output (ATU-III) 10: SSLB2 output (RSPI) 11: TIF6 input (ATU-III)

(2) Port G Control Register 1 (PGCR1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PG7MD[1:0]		PG6MD[1:0]		PG5MD[1:0]		PG4MD[1:0]		PG3MD[1:0]		PG2MD[1:0]		PG1MD[1:0]		PG0MD[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	PG7MD [1:0]	00	R/W	PG7 Mode Control the function of the PG7/TOD13A/SSLB1 pin. 00: PG7 input/output (port) 01: TOD13A output (ATU-III) 10: SSLB1 output (RSPI) 11: Setting prohibited
13, 12	PG6MD [1:0]	00	R/W	PG6 Mode Control the function of the PG6/TOD12A/SSLB0 pin. 00: PG6 input/output (port) 01: TOD12A output (ATU-III) 10: SSLB0 input/output (RSPI) 11: Setting prohibited
11, 10	PG5MD [1:0]	00	R/W	PG5 Mode Control the function of the PG5/TOD11A/SSLA5/SSLC3 pin. 00: PG5 input/output (port) 01: TOD11A output (ATU-III) 10: SSLA5 output (RSPI) 11: SSLC3 output (RSPI)

Bit	Bit Name	Initial Value	R/W	Description
9, 8	PG4MD [1:0]	00	R/W	PG4 Mode Control the function of the PG4/TOD10A/SSLA4/SSLB3 pin. 00: PG4 input/output (port) 01: TOD10A output (ATU-III) 10: SSLA4 output (RSPI) 11: SSLB3 output (RSPI)
7, 6	PG3MD [1:0]	00	R/W	PG3 Mode Control the function of the PG3/TOD03A/SSLA3 pin. 00: PG3 input/output (port) 01: TOD03A output (ATU-III) 10: SSLA3 output (RSPI) 11: Setting prohibited
5, 4	PG2MD [1:0]	00	R/W	PG2 Mode Control the function of the PG2/TOD02A/SSLA2 pin. 00: PG2 input/output (port) 01: TOD02A output (ATU-III) 10: SSLA2 output (RSPI) 11: Setting prohibited
3, 2	PG1MD [1:0]	00	R/W	PG1 Mode Control the function of the PG1/TOD01A/SSLA1 pin. 00: PG1 input/output (port) 01: TOD01A output (ATU-III) 10: SSLA1 output (RSPI) 11: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
1, 0	PG0MD [1:0]	00	R/W	PG0 Mode Control the function of the PG0/TOD00A/SSLA0 pin. 00: PG0 input/output (port) 01: TOD00A output (ATU-III) 10: SSLA0 input/output (RSPI) 11: Setting prohibited

23.1.21 Port H I/O Register (PHIOR)

PHIOR is a 16-bit readable/writable register that sets the I/O direction of the port H pins.

PHIOR is enabled only when the port H pins function as general I/O pins. Otherwise, the set values of this register have no effect on the pin status.

PHIOR is initialized to H'0000 either by power-on reset, hardware standby, or power-on reset by the WDT. However, it is not initialized in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	PH5 IOR	PH4 IOR	PH3 IOR	PH2 IOR	PH1 IOR	PH0 IOR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5 to 0	PH5IOR to PH0IOR	All 0	R/W	The PH5IOR to PH0IOR bits correspond to the PH5 to PH0 pins respectively (the multiplex pin names other than the port names are omitted from pin names). 0: The corresponding pin is set to input. 1: The corresponding pin is set to output.

23.1.22 Port H Control Register (PHCR)

PHCR is a 16-bit readable/writable register that controls the functions of multiplexed pins of port H.

PHCR is initialized to H'0000 either by power-on reset, hardware standby, or power-on reset by the WDT. However, it is not initialized in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	PH5MD[1:0]		PH4MD[1:0]		PH3MD[1:0]		PH2MD[1:0]		PH1MD[1:0]		PH0MD[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11, 10	PH5MD [1:0]	00	R/W	PH5 Mode Control the function of the PH5/TIF5 pin. 00: PH5 input/output (port) 01: Setting prohibited 10: Setting prohibited 11: TIF5 input (ATU-III)
9, 8	PH4MD [1:0]	00	R/W	PH4 Mode Control the function of the PH4/TIF4 pin. 00: PH4 input/output (port) 01: Setting prohibited 10: Setting prohibited 11: TIF4 input (ATU-III)
7, 6	PH3MD [1:0]	00	R/W	PH3 Mode Control the function of the PH3/TIF3 pin. 00: PH3 input/output (port) 01: Setting prohibited 10: Setting prohibited 11: TIF3 input (ATU-III)

Bit	Bit Name	Initial Value	R/W	Description
5, 4	PH2MD [1:0]	00	R/W	PH2 Mode Control the function of the PH2/TIF2A pin. 00: PH2 input/output (port) 01: Setting prohibited 10: Setting prohibited 11: TIF2A input (ATU-III)
3, 2	PH1MD [1:0]	00	R/W	PH1 Mode Control the function of the PH1/ADTRG_B/TIF1A pin. 00: PH1 input/output (port) 01: Setting prohibited 10: ADTRG_B input (ADC) 11: TIF1A input (ATU-III)
1, 0	PH0MD [1:0]	00	R/W	PH0 Mode Control the function of the PH0/ADTRG_A/TIF0A pin. 00: PH0 input/output (port) 01: Setting prohibited 10: ADTRG_A input (ADC) 11: TIF0A input (ATU-III)

23.1.23 Port J I/O Register (PJIOR)

PJIOR is a 16-bit readable/writable register that sets the I/O direction of the port J pins.

PJIOR is enabled only when the port J pins function as general I/O pins. Otherwise, the set values of this register have no effect on the pin status.

PJIOR is initialized to H'0000 either by power-on reset, hardware standby, or power-on reset by the WDT. However, it is not initialized in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	PJ9 IOR	PJ8 IOR	PJ7 IOR	PJ6 IOR	PJ5 IOR	PJ4 IOR	PJ3 IOR	PJ2 IOR	PJ1 IOR	PJ0 IOR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 0	PJ9IOR to PJ0IOR	All 0	R/W	The PJ9IOR to PJ0IOR bits correspond to the PJ9 to PJ0 pins respectively (the multiplex pin names other than the port names are omitted from pin names). 0: The corresponding pin is set to input. 1: The corresponding pin is set to output.

23.1.24 Port J Control Registers 1 and 2 (PJCR1 and PJCR2)

PJCR1 and PJCR2 are 16-bit readable/writable registers that control the functions of multiplexed pins of port J.

PJCR1 and PJCR2 are initialized to H'0000 either by power-on reset, hardware standby, or power-on reset by the WDT. However, they are not initialized in sleep mode.

(1) Port J Control Register 2 (PJCR2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	PJ9 MD	-	PJ8 MD
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	PJ9MD	0	R/W	PJ9 Mode Controls the function of the PJ9/RxD_B pin. 0: PJ9 input/output (port) 1: RxD_B input (SCI)
1	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
0	PJ8MD	0	R/W	PJ8 Mode Controls the function of the PJ8/TxD_B pin. 0: PJ8 input/output (port) 1: TxD_B output (SCI)

(2) Port J Control Register 1 (PJCR1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PJ7MD[1:0]		-	PJ6 MD	-	PJ5 MD	PJ4MD[1:0]		PJ3MD[1:0]		PJ2MD[1:0]		PJ1MD[1:0]		PJ0MD[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	PJ7MD [1:0]	00	R/W	PJ7 Mode Control the function of the PJ7/SCK_B/ADEND_A/TIJ1 pin. 00: PJ7 input/output (port) 01: SCK_B input/output (SCI) 10: ADEND_A output (ADC) 11: TIJ1 input (ATU-III)
13	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
12	PJ6MD	0	R/W	PJ6 Mode Controls the function of the PJ6/RxD_A pin. 0: PJ6 input/output (port) 1: RxD_A input (SCI)
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10	PJ5MD	0	R/W	PJ5 Mode Controls the function of the PJ5/TxD_A pin. 0: PJ5 input/output (port) 1: TxD_A output (SCI)

Bit	Bit Name	Initial Value	R/W	Description
9, 8	PJ4MD [1:0]	00	R/W	PJ4 Mode Control the function of the PJ4/SCK_A/ADEND_B/TIJ0 pin. 00: PJ4 input/output (port) 01: SCK_A input/output (SCI) 10: ADEND_B output (ADC) 11: TIJ0 input (ATU-III)
7, 6	PJ3MD [1:0]	00	R/W	PJ3 Mode Control the function of the PJ3/RxD_A/CRx_C/CRx_A&CRx_B&CRx_C pin. 00: PJ3 input/output (port) 01: RxD_A input (SCI) 10: CRx_C input (RCAN-TL1) 11: CRx_A&CRx_B&CRx_C input (RCAN-TL1)
5, 4	PJ2MD [1:0]	00	R/W	PJ2 Mode Control the function of the PJ2/TxD_A/CTx_C/CTx_A&CTx_B&CTx_C pin. 00: PJ2 input/output (port) 01: TxD_A output (SCI) 10: CTx_C output (RCAN-TL1) 11: CTx_A&CTx_B&CTx_C output (RCAN-TL1)
3, 2	PJ1MD [1:0]	00	R/W	PJ1 Mode Control the function of the PJ1/RxD_A/CRx_A/CRx_A&CRx_B pin. 00: PJ1 input/output (port) 01: RxD_A input (SCI) 10: CRx_A input (RCAN-TL1) 11: CRx_A&CRx_B input (RCAN-TL1)

Bit	Bit Name	Initial Value	R/W	Description
1, 0	PJ0MD [1:0]	00	R/W	PJ0 Mode Control the function of the PJ0/TxD_A/CTx_A/CTx_A&CTx_B pin. 00: PJ0 input/output (port) 01: TxD_A output (SCI) 10: CTx_A output (RCAN-TL1) 11: CTx_A&CTx_B output (RCAN-TL1)

23.1.25 Port J Control Registers 1A to 3A (PJCR1A to PJCR3A)

PJCR1A and PJCR3A are 16-bit readable/writable registers that control the functions of multiplexed pins of port J.

PJCR1A and PJCR3A are initialized to H'0000 either by power-on reset, hardware standby, or power-on reset by the WDT. However, they are not initialized in sleep mode.

(1) Port J Control Register 3A (PJCR3A)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	PJ9AMD[1:0]		-	-	PJ8AMD[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5, 4	PJ9AMD [1:0]	00	R/W	PJ9 Mode Controls the function of the PJ9/CRx_D pin. 00: PJ9 input/output (port) 01: Setting prohibited 10: Setting prohibited 11: CRx_D input (RCAN) Note: This setting is valid only when PJ9MD = 0.
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	PJ8AMD [1:0]	00	R/W	PJ8 Mode Controls the function of the PJ8/CTx_D pin. 00: PJ8 input/output (port) 01: Setting prohibited 10: Setting prohibited 11: CTx_D output (RCAN) Note: This setting is valid only when PJ8MD = 0.

(2) Port J Control Register 2A (PJCR2A)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PJ7AMD[1:0]		-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13, 12	PJ7AMD [1:0]	00	R/W	PJ7 Mode Control the function of the PJ7/FREN_A pin. 00: PJ7 input/output (port) 01: Setting prohibited 10: Setting prohibited 11: FREN_A output (FlexRay) Note: This setting is valid only when PJ7MD[1:0] = 00.
11 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

(3) Port J Control Register 1A (PJCR1A)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PJ3AMD[1:0]		-	-	PJ2AMD[1:0]		-	-	PJ1AMD[1:0]		-	-	PJ0AMD[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13, 12	PJ3AMD [1:0]	00	R/W	PJ3 Mode Control the function of the PJ3/RxD_B/CRx_C&CRx_D pin. 00: PJ3 input/output (port) 01: Setting prohibited 10: RxD_B input (SCI) 11: CRx_C&CRx_D input (RCAN) Note: This setting is valid only when PJ3MD[1:0] = 00.
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	PJ2AMD [1:0]	00	R/W	PJ2 Mode Control the function of the PJ2/TxD_B/CTx_C&CTx_D pin. 00: PJ2 input/output (port) 01: Setting prohibited 10: TxD_B output (SCI) 11: CTx_C&CTx_D output (RCAN) Note: This setting is valid only when PJ2MD[1:0] = 00.
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
5, 4	PJ1AMD [1:0]	00	R/W	PJ1 Mode Control the function of the PJ1/RxD_B/FRRxD_A pin. 00: PJ1 input/output (port) 01: Setting prohibited 10: RxD_B input (SCI) 11: FRRxD_A input (FlexRay) Note: This setting is valid only when PJ1MD[1:0] = 00.
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	PJ0AMD [1:0]	00	R/W	PJ0 Mode Control the function of the PJ0/TxD_B/FRTxD_A pin. 00: PJ0 input/output (port) 01: Setting prohibited 10: TxD_B output (SCI) 11: FRTxD_A output (FlexRay) Note: This setting is valid only when PJ0MD[1:0] = 00.

23.1.26 Port K I/O Register (PKIOR)

PKIOR is a 16-bit readable/writable register that sets the I/O direction of the port K pins.

PKIOR is enabled only when the port K pins function as general I/O pins. Otherwise, the set values of this register have no effect on the pin status.

PKIOR is initialized to H'0000 either by power-on reset, hardware standby, or power-on reset by the WDT. However, it is not initialized in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	PK11 IOR	PK10 IOR	PK9 IOR	PK8 IOR	PK7 IOR	PK6 IOR	PK5 IOR	PK4 IOR	PK3 IOR	PK2 IOR	PK1 IOR	PK0 IOR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 0	PK11IOR to PK0IOR	All 0	R/W	The PK11IOR to PK0IOR bits correspond to the PK11 to PK0 pins respectively (the multiplex pin names other than the port names are omitted from pin names). 0: The corresponding pin is set to input. 1: The corresponding pin is set to output.

23.1.27 Port K Control Registers 1 and 2 (PKCR1 and PKCR2)

PKCR1 and PKCR2 are 16-bit readable/writable registers that control the functions of multiplexed pins of port K.

PKCR1 and PKCR2 are initialized to H'0000 either by power-on reset, hardware standby, or power-on reset by the WDT. However, they are not initialized in sleep mode.

(1) Port K Control Register 2 (PKCR2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	PK11MD[1:0]		PK10MD[1:0]		PK9MD[1:0]		-	PK8 MD
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7, 6	PK11MD [1:0]	00	R/W	PK11 Mode Control the function of the PK11/MISOC pin. 00: PK11 input/output (port) 01: Setting prohibited 10: MISOC input/output (RSPI) 11: Setting prohibited
5, 4	PK10MD [1:0]	00	R/W	PK10 Mode Control the function of the PK10/MOSIC pin. 00: PK10 input/output (port) 01: Setting prohibited 10: MOSIC input/output (RSPI) 11: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
3, 2	PK9MD [1:0]	00	R/W	PK9 Mode Control the function of the PK9/RSPCKC pin. 00: PK9 input/output (port) 01: Setting prohibited 10: RSPCKC input/output (RSPI) 11: Setting prohibited
1	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
0	PK8MD	0	R/W	PK8 Mode Controls the function of the PK8/RxD_E pin. 0: PK8 input/output (port) 1: RxD_E input (SCI)

(2) Port K Control Register 1 (PKCR1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PK7 MD	-	PK6 MD	PK5MD[1:0]		PK4MD[1:0]		PK3MD[1:0]		PK2MD[1:0]		PK1MD[1:0]		PK0MD[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14	PK7MD	0	R/W	PK7 Mode Controls the function of the PK7/TxD_E pin. 0: PK7 input/output (port) 1: TxD_E output (SCI)
13	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
12	PK6MD	0	R/W	PK6 Mode Controls the function of the PK6/SCK_E pin. 0: PK6 input/output (port) 1: SCK_E input/output (SCI)
11, 10	PK5MD [1:0]	00	R/W	PK5 Mode Control the function of the PK5/RxD_D/MISOB pin. 00: PK5 input/output (port) 01: RxD_D input (SCI) 10: MISOB input/output (RSPI) 11: Setting prohibited
9, 8	PK4MD [1:0]	00	R/W	PK4 Mode Control the function of the PK4/TxD_D/MOSIB pin. 00: PK4 input/output (port) 01: TxD_D output (SCI) 10: MOSIB input/output (RSPI) 11: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
7, 6	PK3MD [1:0]	00	R/W	PK3 Mode Control the function of the PK3/SCK_D/RSPCKB pin. 00: PK3 input/output (port) 01: SCK_D input/output (SCI) 10: RSPCKB input/output (RSPI) 11: Setting prohibited
5, 4	PK2MD [1:0]	00	R/W	PK2 Mode Control the function of the PK2/RxD_C/MISOA pin. 00: PK2 input/output (port) 01: RxD_C input (SCI) 10: MISOA input/output (RSPI) 11: Setting prohibited
3, 2	PK1MD [1:0]	00	R/W	PK1 Mode Control the function of the PK1/TxD_C/MOSIA pin. 00: PK1 input/output (port) 01: TxD_C output (SCI) 10: MOSIA input/output (RSPI) 11: Setting prohibited
1, 0	PK0MD [1:0]	00	R/W	PK0 Mode Control the function of the PK0/SCK_C/RSPCKA/$\overline{\text{UBCTRG}}$ pin. 00: PK0 input/output (port) 01: SCK_C input/output (ATU-III) 10: RSPCKA input/output (RSPI) 11: $\overline{\text{UBCTRG}}$ output (UBC)

23.1.28 Port L I/O Register (PLIOR)

PLIOR is a 16-bit readable/writable register that sets the I/O direction of the port L pins.

PLIOR is enabled only when the port L pins function as general I/O pins. Otherwise, the set values of this register have no effect on the pin status.

PLIOR is initialized to H'0000 either by power-on reset, hardware standby, or power-on reset by the WDT. However, it is not initialized in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	PL8 IOR	PL7 IOR	PL6 IOR	PL5 IOR	PL4 IOR	PL3 IOR	PL2 IOR	PL1 IOR	PL0 IOR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8 to 0	PL8IOR to PL0IOR	All 0	R/W	The PL8IOR to PL0IOR bits correspond to the PL8 to PL0 pins respectively (the multiplex pin names other than the port names are omitted from pin names). 0: The corresponding pin is set to input. 1: The corresponding pin is set to output.

23.1.29 Port L Control Registers 1 and 2 (PLCR1 and PLCR2)

PLCR1 and PLCR2 are 16-bit readable/writable registers that control the functions of multiplexed pins of port L.

PLCR1 and PLCR2 are initialized to H'0000 either by power-on reset, hardware standby, or power-on reset by the WDT. However, they are not initialized in sleep mode.

(1) Port L Control Register 2 (PLCR2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	PL8 MD
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	PL8MD	0	R/W	PL8 Mode Controls the function of the PL8/TOE33 pin. 0: PL8 input/output (port) 1: TOE33 output (ATU-III)

(2) Port L Control Register 1 (PLCR1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PL7MD[1:0]		PL6MD[1:0]		PL5MD[1:0]		PL4MD[1:0]		PL3MD[1:0]		PL2MD[1:0]		PL1MD[1:0]		PL0MD[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	PL7MD [1:0]	00	R/W	PL7 Mode Control the function of the PL7/TOE32/$\overline{\text{IRQ7}}$ pin. 00: PL7 input/output (port) 01: TOE32 output (ATU-III) 10: $\overline{\text{IRQ7}}$ input (INTC) 11: Setting prohibited
13, 12	PL6MD [1:0]	00	R/W	PL6 Mode Control the function of the PL6/TOE31/$\overline{\text{IRQ6}}$ pin. 00: PL6 input/output (port) 01: TOE31 output (ATU-III) 10: $\overline{\text{IRQ6}}$ input (INTC) 11: Setting prohibited
11, 10	PL5MD [1:0]	00	R/W	PL5 Mode Control the function of the PL5/TOE30/$\overline{\text{IRQ5}}$ pin. 00: PL5 input/output (port) 01: TOE30 output (ATU-III) 10: $\overline{\text{IRQ5}}$ input (INTC) 11: Setting prohibited
9, 8	PL4MD [1:0]	00	R/W	PL4 Mode Control the function of the PL4/TOE23/$\overline{\text{IRQ4}}$ pin. 00: PL4 input/output (port) 01: TOE23 output (ATU-III) 10: $\overline{\text{IRQ4}}$ input (INTC) 11: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
7, 6	PL3MD [1:0]	00	R/W	PL3 Mode Control the function of the PL3/TOE22/$\overline{\text{IRQ3}}$ pin. 00: PL3 input/output (port) 01: TOE22 output (ATU-III) 10: $\overline{\text{IRQ3}}$ input (INTC) 11: Setting prohibited
5, 4	PL2MD [1:0]	00	R/W	PL2 Mode Control the function of the PL2/TOE21/$\overline{\text{IRQ2}}$ pin. 00: PL2 input/output (port) 01: TOE21 output (ATU-III) 10: $\overline{\text{IRQ2}}$ input (INTC) 11: Setting prohibited
3, 2	PL1MD [1:0]	00	R/W	PL1 Mode Control the function of the PL1/TOE20/$\overline{\text{IRQ1}}$/$\overline{\text{POD}}$ pin. 00: PL1 input/output (port) 01: TOE20 output (ATU-III) 10: $\overline{\text{IRQ1}}$ input (INTC) 11: $\overline{\text{POD}}$ input (port)
1, 0	PL0MD [1:0]	00	R/W	PL0 Mode Control the function of the PL0/ $\overline{\text{IRQ0}}$ pin. 00: PL0 input/output (port) 01: Setting prohibited 10: $\overline{\text{IRQ0}}$ input (INTC) 11: Setting prohibited

Section 24 I/O Ports

24.1 Overview

This LSI provides 11 ports: ports A, B, C, D, E, F, G, H, J, K, and L.

Ports A, C, F, and G are 16-bit I/O ports. Port B is a 15-bit I/O port. Ports D and E are 14-bit I/O ports. Port H is a 6-bit I/O port. Port J is a 10-bit I/O port. Port K is a 12-bit I/O port. Port L is a 9-bit I/O port.

Each port pin which is a general I/O port also has other multiplexed functions. The function of the multiplexed pin is selected by the pin function controller (PFC).

Each port has a data register to store pin data and a port register to read the pin state.

In addition, each of the ports A, B, C, D, E, F, G, J, K, and L incorporates an inverting register to output the inverted set value.

Port G can detect an edge input on a pin. Port G has an edge selecting register to select an edge.

Ports B, C, E, F, G, J, and K can set the driving ability of pins. Each of these ports has a driving ability setting register for this purpose.

Ports B, C, F, J, and K can set the pull-down for pins. Each of these ports has a pin state setting register for this purpose.

In addition, this LSI provides the CK control register (CKCR) to enable or disable the CK pin output.

24.2 Register Descriptions

The I/O port registers are listed in table 24.1.

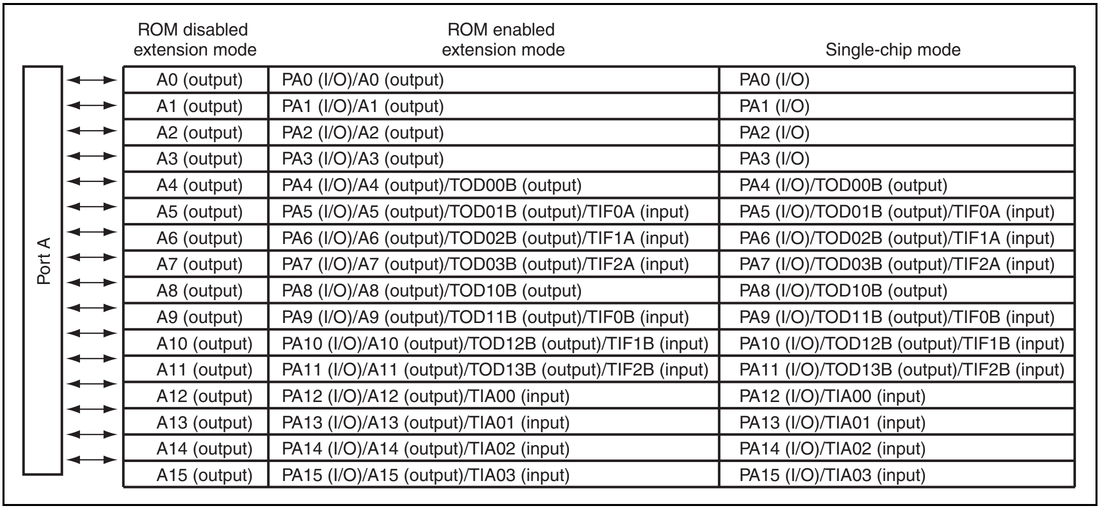
Table 24.1 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port A data register	PADR	R/W	H'0000	H'FFFE3802	8, 16
Port A port register	PAPR	R	Pin state	H'FFFE381E	8, 16
Port A inverting register	PAIR	R/W	H'0000	H'FFFE3818	8, 16
Port B data register	PBDR	R/W	H'0000	H'FFFE3882	8, 16
Port B port register	PBPR	R	Pin state	H'FFFE389E	8, 16
Port B inverting register	PBIR	R/W	H'0000	H'FFFE3898	8, 16, 32
Port B driving ability setting register	PBDSR	R/W	H'0000	H'FFFE389A	8, 16
Port B pin state setting register	PBPSR	R/W	H'0000	H'FFFE389C	8, 16, 32
Port C data register	PCDR	R/W	H'0000	H'FFFE3902	8, 16
Port C port register	PCPR	R	Pin state	H'FFFE391E	8, 16
Port C inverting register	PCIR	R/W	H'0000	H'FFFE3918	8, 16, 32
Port C driving ability setting register	PCDSR	R/W	H'0000	H'FFFE391A	8, 16
Port C pin state setting register	PCPSR	R/W	H'0000	H'FFFE391C	8, 16, 32
Port D data register	PDDR	R/W	H'0000	H'FFFFC800	8, 16, 32
Port D port register	PDPR	R	Pin state	H'FFFFC802	8, 16
Port D inverting register	PDIR	R/W	H'0000	H'FFFFC804	8, 16
Port E data register	PEDR	R/W	H'0000	H'FFFFC810	8, 16, 32
Port E port register	PEPR	R	Pin state	H'FFFFC812	8, 16
Port E inverting register	PEIR	R/W	H'0000	H'FFFFC814	8, 16, 32
Port E driving ability setting register	PEDSR	R/W	H'0000	H'FFFFC816	8, 16
Port F data register	PFDR	R/W	H'0000	H'FFFFC820	8, 16, 32
Port F port register	PFPR	R	Pin state	H'FFFFC822	8, 16
Port F inverting register	PFIR	R/W	H'0000	H'FFFFC824	8, 16, 32

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port F driving ability setting register	PFDSR	R/W	H'0000	H'FFFFC826	8, 16
Port F pin state setting register	PFPSR	R/W	H'0000	H'FFFFC828	8, 16, 32
Port G data register	PGDR	R/W	H'0000	H'FFFFC830	8, 16, 32
Port G port register	PGPR	R	Pin state	H'FFFFC832	8, 16
Port G inverting register	PGIR	R/W	H'0000	H'FFFFC834	8, 16, 32
Port G driving ability setting register	PGDSR	R/W	H'0000	H'FFFFC836	8, 16
Port G edge selecting register	PGER	R/W	H'0000	H'FFFFC838	8, 16
Port H data register	PHDR	R/W	H'0000	H'FFFFC850	8, 16, 32
Port H port register	PHPR	R	Pin state	H'FFFFC852	8, 16
Port J data register	PJDR	R/W	H'0000	H'FFFFC860	8, 16, 32
Port J port register	PJPR	R	Pin state	H'FFFFC862	8, 16
Port J inverting register	PJIR	R/W	H'0000	H'FFFFC864	8, 16, 32
Port J driving ability setting register	PJDSR	R/W	H'0000	H'FFFFC866	8, 16
Port J pin state setting register	PJPSR	R/W	H'0000	H'FFFFC868	8, 16
Port K data register	PKDR	R/W	H'0000	H'FFFFC880	8, 16, 32
Port K port register	PKPR	R	Pin state	H'FFFFC882	8, 16
Port K inverting register	PKIR	R/W	H'0000	H'FFFFC884	8, 16, 32
Port K driving ability setting register	PKDSR	R/W	H'0000	H'FFFFC886	8, 16
Port K pin state setting register	PKPSR	R/W	H'0000	H'FFFFC888	8, 16
Port L data register	PLDR	R/W	H'0000	H'FFFFC8A0	8, 16, 32
Port L port register	PLPR	R	Pin state	H'FFFFC8A2	8, 16
Port L inverting register	PLIR	R/W	H'0000	H'FFFFC8A4	8, 16
CK control register	CKCR	R/W	H'0000	H'FFFFC920	8, 16

24.3 Port A

Port A is an input/output port with the 16 pins shown in figure 24.1.



ROM disabled extension mode	ROM enabled extension mode	Single-chip mode
A0 (output)	PA0 (I/O)/A0 (output)	PA0 (I/O)
A1 (output)	PA1 (I/O)/A1 (output)	PA1 (I/O)
A2 (output)	PA2 (I/O)/A2 (output)	PA2 (I/O)
A3 (output)	PA3 (I/O)/A3 (output)	PA3 (I/O)
A4 (output)	PA4 (I/O)/A4 (output)/TOD00B (output)	PA4 (I/O)/TOD00B (output)
A5 (output)	PA5 (I/O)/A5 (output)/TOD01B (output)/TIF0A (input)	PA5 (I/O)/TOD01B (output)/TIF0A (input)
A6 (output)	PA6 (I/O)/A6 (output)/TOD02B (output)/TIF1A (input)	PA6 (I/O)/TOD02B (output)/TIF1A (input)
A7 (output)	PA7 (I/O)/A7 (output)/TOD03B (output)/TIF2A (input)	PA7 (I/O)/TOD03B (output)/TIF2A (input)
A8 (output)	PA8 (I/O)/A8 (output)/TOD10B (output)	PA8 (I/O)/TOD10B (output)
A9 (output)	PA9 (I/O)/A9 (output)/TOD11B (output)/TIF0B (input)	PA9 (I/O)/TOD11B (output)/TIF0B (input)
A10 (output)	PA10 (I/O)/A10 (output)/TOD12B (output)/TIF1B (input)	PA10 (I/O)/TOD12B (output)/TIF1B (input)
A11 (output)	PA11 (I/O)/A11 (output)/TOD13B (output)/TIF2B (input)	PA11 (I/O)/TOD13B (output)/TIF2B (input)
A12 (output)	PA12 (I/O)/A12 (output)/TIA00 (input)	PA12 (I/O)/TIA00 (input)
A13 (output)	PA13 (I/O)/A13 (output)/TIA01 (input)	PA13 (I/O)/TIA01 (input)
A14 (output)	PA14 (I/O)/A14 (output)/TIA02 (input)	PA14 (I/O)/TIA02 (input)
A15 (output)	PA15 (I/O)/A15 (output)/TIA03 (input)	PA15 (I/O)/TIA03 (input)

Figure 24.1 Port A

24.3.1 Port A Data Register (PADR)

PADR is a 16-bit readable/writable register that stores port A data. The PA15DR to PA0DR bits correspond to the PA15/A15/TIA03 to PA0/A0 pins, respectively.

When a pin functions as a general output, a value written to a corresponding bit in this register is output on the pin. The bit value can be read directly regardless of the pin state by reading the bit in this register.

When this register is read while a pin functions as a general input, the pin state not a register value can be read. When data is written to this register while the pin functions as a general input, data can be written but it does not affect the pin state. Table 24.2 summarizes the PADR read and write operations.

PADR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PA15DR	PA14DR	PA13DR	PA12DR	PA11DR	PA10DR	PA9DR	PA8DR	PA7DR	PA6DR	PA5DR	PA4DR	PA3DR	PA2DR	PA1DR	PA0DR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Descriptions
15	PA15DR	0	R/W	Refer to table 24.2.
14	PA14DR	0	R/W	
13	PA13DR	0	R/W	
12	PA12DR	0	R/W	
11	PA11DR	0	R/W	
10	PA10DR	0	R/W	
9	PA9DR	0	R/W	
8	PA8DR	0	R/W	
7	PA7DR	0	R/W	
6	PA6DR	0	R/W	
5	PA5DR	0	R/W	
4	PA4DR	0	R/W	
3	PA3DR	0	R/W	
2	PA2DR	0	R/W	
1	PA1DR	0	R/W	
0	PA0DR	0	R/W	

Table 24.2 Read and Write Operations of Port A Data Register (PADR)

- Bits 15 to 0 in the PADR register

PAIOR	Pin Function	Read	Write
0	General input	Pin state	Data can be written to PADR but data does not affect the pin state.
	Other than general input	Pin state	Data can be written to PADR but data does not affect the pin state.
1	General output	PADR value	Data written to PADR is output on the pin.
	Other than general output	PADR value	Data can be written to PADR but data does not affect the pin state.

24.3.2 Port A Port Register (PAPR)

PAPR is a 16-bit read-only register that always stores port A pin states. This register cannot be directly written by the CPU. The PA15PR to PA0PR bits correspond to the PA15/A15/TIA03 to PA0/A0 pins, respectively.

When PAPR is read, the pin state can be read.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PA15 PR	PA14 PR	PA13 PR	PA12 PR	PA11 PR	PA10 PR	PA9PR	PA8PR	PA7PR	PA6PR	PA5PR	PA4PR	PA3PR	PA2PR	PA1PR	PA0PR
Initial value:	PA15	PA14	PA13	PA12	PA11	PA10	PA9	PA8	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Descriptions
15	PA15PR	Pin state	R	When these bits are read, the pin states can be read. These bits cannot be modified.
14	PA14PR	Pin state	R	
13	PA13PR	Pin state	R	
12	PA12PR	Pin state	R	
11	PA11PR	Pin state	R	
10	PA10PR	Pin state	R	
9	PA9PR	Pin state	R	
8	PA8PR	Pin state	R	
7	PA7PR	Pin state	R	
6	PA6PR	Pin state	R	
5	PA5PR	Pin state	R	
4	PA4PR	Pin state	R	
3	PA3PR	Pin state	R	
2	PA2PR	Pin state	R	
1	PA1PR	Pin state	R	
0	PA0PR	Pin state	R	

24.3.3 Port A Inverting Register (PAIR)

PAIR is a 16-bit readable/writable register that enables or disables the inverting function for port A. Bits PA11IR to PA4IR correspond to pins PA11/A11/TOD13B/TIF2B to PA4/A4/TOD00B, respectively. The PAIR setting is ignored when the corresponding pin function is specified as a bus function.

Setting the specific bits of PAIR to 1 reverts the output values on their corresponding pins.

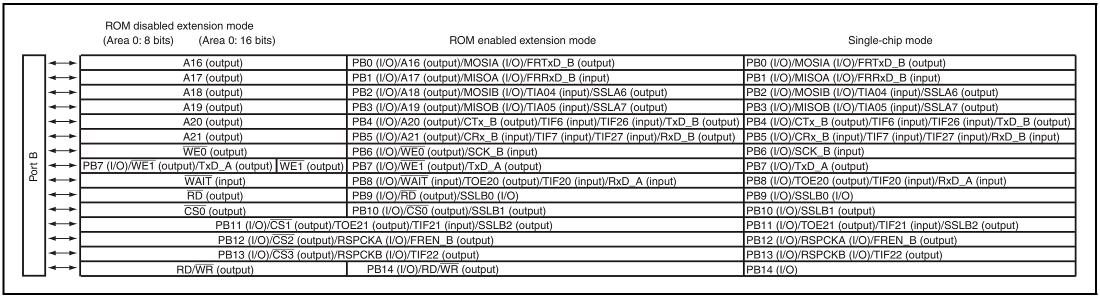
PAIR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	PA11IR	PA10IR	PA9IR	PA8IR	PA7IR	PA6IR	PA5IR	PA4IR	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Descriptions
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11	PA11IR	0	R/W	These bits set whether or not the set value is inverted when it is output on a corresponding pin. 0: The set value is not inverted when output 1: The set value is inverted when output
10	PA10IR	0	R/W	
9	PA9IR	0	R/W	
8	PA8IR	0	R/W	
7	PA7IR	0	R/W	
6	PA6IR	0	R/W	
5	PA5IR	0	R/W	
4	PA4IR	0	R/W	
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

24.4 Port B

Port B is an input/output port with the 15 pins shown in figure 24.2.



ROM disabled extension mode			
(Area 0: 8 bits)	(Area 0: 16 bits)	ROM enabled extension mode	Single-chip mode
A16 (output)		PB0 (I/O)/A16 (output)/MOSIA (I/O)/FRTxD_B (output)	PB0 (I/O)/MOSIA (I/O)/FRTxD_B (output)
A17 (output)		PB1 (I/O)/A17 (output)/MISOA (I/O)/FRRxD_B (input)	PB1 (I/O)/MISOA (I/O)/FRRxD_B (input)
A18 (output)		PB2 (I/O)/A18 (output)/MOSIB (I/O)/TIA04 (input)/SSLA6 (output)	PB2 (I/O)/MOSIB (I/O)/TIA04 (input)/SSLA6 (output)
A19 (output)		PB3 (I/O)/A19 (output)/MISOB (I/O)/TIA05 (input)/SSLA7 (output)	PB3 (I/O)/MISOB (I/O)/TIA05 (input)/SSLA7 (output)
A20 (output)		PB4 (I/O)/A20 (output)/CTx_B (output)/TIF6 (input)/TIF26 (input)/TxD_B (output)	PB4 (I/O)/CTx_B (output)/TIF6 (input)/TIF26 (input)/TxD_B (output)
A21 (output)		PB5 (I/O)/A21 (output)/CRx_B (input)/TIF7 (input)/TIF27 (input)/RxD_B (output)	PB5 (I/O)/CRx_B (input)/TIF7 (input)/TIF27 (input)/RxD_B (input)
$\overline{\text{WE0}}$ (output)		PB6 (I/O)/$\overline{\text{WE0}}$ (output)/SCK_B (input)	PB6 (I/O)/SCK_B (input)
PB7 (I/O)/$\overline{\text{WE1}}$ (output)/TxD_A (output)	$\overline{\text{WE1}}$ (output)	PB7 (I/O)/$\overline{\text{WE1}}$ (output)/TxD_A (output)	PB7 (I/O)/TxD_A (output)
$\overline{\text{WAIT}}$ (input)		PB8 (I/O)/$\overline{\text{WAIT}}$ (input)/TOE20 (output)/TIF20 (input)/RxD_A (input)	PB8 (I/O)/TOE20 (output)/TIF20 (input)/RxD_A (input)
$\overline{\text{RD}}$ (output)		PB9 (I/O)/$\overline{\text{RD}}$ (output)/SSLB0 (I/O)	PB9 (I/O)/SSLB0 (I/O)
$\overline{\text{CS0}}$ (output)		PB10 (I/O)/$\overline{\text{CS0}}$ (output)/SSLB1 (output)	PB10 (I/O)/SSLB1 (output)
PB11 (I/O)/$\overline{\text{CS1}}$ (output)/TOE21 (output)/TIF21 (input)/SSLB2 (output)			PB11 (I/O)/TOE21 (output)/TIF21 (input)/SSLB2 (output)
PB12 (I/O)/$\overline{\text{CS2}}$ (output)/RSPCKA (I/O)/FREN_B (output)			PB12 (I/O)/RSPCKA (I/O)/FREN_B (output)
PB13 (I/O)/$\overline{\text{CS3}}$ (output)/RSPCKB (I/O)/TIF22 (output)			PB13 (I/O)/RSPCKB (I/O)/TIF22 (output)
RD/$\overline{\text{WR}}$ (output)		PB14 (I/O)/RD/$\overline{\text{WR}}$ (output)	PB14 (I/O)

Figure 24.2 Port B

24.4.1 Port B Data Register (PBDR)

PBDR is a 16-bit readable/writable register that stores port B data. The PB14DR to PB0DR bits correspond to the PB14/RD/$\overline{\text{WR}}$ to PB0/A16/MOSIA/FRTxD_B pins, respectively.

When a pin functions as a general output, a value written to a corresponding bit in this register is output on the pin. The bit value can be read directly regardless of the pin state by reading the bit in this register.

When this register is read while a pin functions as a general input, the pin state not a register value can be read. When data is written to this register while the pin functions as a general input, data can be written but it does not affect the pin state. Table 24.3 summarizes the PBDR read and write operations.

PBDR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PB14 DR	PB13 DR	PB12 DR	PB11 DR	PB10 DR	PB9DR	PB8DR	PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Descriptions
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14	PB14DR	0	R/W	Refer to table 24.3.
13	PB13DR	0	R/W	
12	PB12DR	0	R/W	
11	PB11DR	0	R/W	
10	PB10DR	0	R/W	
9	PB9DR	0	R/W	
8	PB8DR	0	R/W	
7	PB7DR	0	R/W	
6	PB6DR	0	R/W	
5	PB5DR	0	R/W	
4	PB4DR	0	R/W	
3	PB3DR	0	R/W	
2	PB2DR	0	R/W	
1	PB1DR	0	R/W	
0	PB0DR	0	R/W	

Table 24.3 Read and Write Operations of Port B Data Register (PBDR)

- Bits 15 to 0 in PBDR

PBIOR	Pin Function	Read	Write
0	General input	Pin state	Data can be written to PBDR but data does not affect the pin state.
	Other than general input	Pin state	Data can be written to PBDR but data does not affect the pin state.
1	General output	PBDR value	Data written to PBDR is output on the pin.
	Other than general output	PBDR value	Data can be written to PBDR but data does not affect the pin state.

24.4.2 Port B Port Register (PBPR)

PBPR is a 16-bit read-only register that always stores port B pin states. This register cannot be directly written by the CPU. The PB14PR to PB0PR bits correspond to the PB14/RD/$\overline{\text{WR}}$ to PB0/A16/MOSIA/FRTxD_B pins, respectively.

When PBPR is read, the pin state can be read.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PB14 PR	PB13 PR	PB12 PR	PB11 PR	PB10 PR	PB9PR	PB8PR	PB7PR	PB6PR	PB5PR	PB4PR	PB3PR	PB2PR	PB1PR	PB0PR
Initial value:	0	PB14	PB13	PB12	PB11	PB10	PB9	PB8	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Descriptions
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14	PB14PR	Pin state	R	When these bits are read, the pin states can be read. These bits cannot be modified.
13	PB13PR	Pin state	R	
12	PB12PR	Pin state	R	
11	PB11PR	Pin state	R	
10	PB10PR	Pin state	R	
9	PB9PR	Pin state	R	
8	PB8PR	Pin state	R	
7	PB7PR	Pin state	R	
6	PB6PR	Pin state	R	
5	PB5PR	Pin state	R	
4	PB4PR	Pin state	R	
3	PB3PR	Pin state	R	
2	PB2PR	Pin state	R	
1	PB1PR	Pin state	R	
0	PB0PR	Pin state	R	

24.4.3 Port B Inverting Register (PBIR)

PBIR is a 16-bit readable/writable register that enables or disables the inverting function for port B. Bits PB13IR to PB6IR and PB4IR to PB0IR correspond to pins PB13/$\overline{\text{CS3}}$/RSPCKB/TIF22 to PB6/$\overline{\text{WE0}}$/SCK_B and PB4/A20/CTx_B/TIF6/TIF26/TxD_B to PB0/A16/MOSIA/FRTxD_B, respectively. The PBIR setting is ignored when the corresponding pin function is specified as a bus function (A16 to A20, $\overline{\text{WE0}}$, $\overline{\text{WE1}}$, $\overline{\text{WAIT}}$, $\overline{\text{RD}}$, or $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$).

Setting the specific bits of PBIR to 1 reverts the output values on their corresponding pins.

PBIR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PB13IR	PB12IR	PB11IR	PB10IR	PB9IR	PB8IR	PB7IR	PB6IR	-	PB4IR	PB3IR	PB2IR	PB1IR	PB0IR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Descriptions
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	PB13IR	0	R/W	These bits set whether or not the set value is inverted when it is output on a corresponding pin. 0: The set value is not inverted when output 1: The set value is inverted when output
12	PB12IR	0	R/W	
11	PB11IR	0	R/W	
10	PB10IR	0	R/W	
9	PB9IR	0	R/W	
8	PB8IR	0	R/W	
7	PB7IR	0	R/W	
6	PB6IR	0	R/W	
5	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Descriptions
4	PB4IR	0	R/W	These bits set whether or not the set value is inverted when it is output on a corresponding pin. 0: The set value is not inverted when output 1: The set value is inverted when output
3	PB3IR	0	R/W	
2	PB2IR	0	R/W	
1	PB1IR	0	R/W	
0	PB0IR	0	R/W	

24.4.4 Port B Driving Ability Setting Register (PBDSR)

PBDSR is a 16-bit readable/writable register that selects the driving ability for port B. Bits PB13DSR to PB9DSR, PB7DSR, PB6DSR, and PB4DSR to PB0DSR correspond to pins PB13/$\overline{\text{CS3}}$/RSPCKB/TIF22 to PB9/$\overline{\text{RD}}$/SSLB0, PB7/$\overline{\text{WE1}}$/TxD_A, PB6/$\overline{\text{WE0}}$/SCK_B, and PB4/A20/CTx_B/TIF6/TIF26/TxD_B to PB0/A16/MOSIA/FRTxD_B, respectively. The PBDSR setting is ignored when the corresponding pin function is specified as bus function (A16 to A20, $\overline{\text{WE0}}$, $\overline{\text{WE1}}$, $\overline{\text{RD}}$, or $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$).

Setting the specific bits of PBDSR to 1 increases the driving ability of their corresponding pins.

PBDSR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PB13 DSR	PB12 DSR	PB11 DSR	PB10 DSR	PB9 DSR	-	PB7 DSR	PB6 DSR	-	PB4 DSR	PB3 DSR	PB2 DSR	PB1 DSR	PB0 DSR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Descriptions
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	PB13DSR	0	R/W	When these bits are set to 1, the driving ability of the corresponding pin is higher than normal. 0: Normal driving ability (slow slew rate) 1: High driving ability (fast slew rate) Note: For the characteristics of the driving ability, refer to section 34.3.14, Output Slew Rate.
12	PB12DSR	0	R/W	
11	PB11DSR	0	R/W	
10	PB10DSR	0	R/W	
9	PB9DSR	0	R/W	

Bit	Bit Name	Initial Value	R/W	Descriptions
8	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
7	PB7DSR	0	R/W	When these bits are set to 1, the driving ability of the corresponding pin is higher than normal. 0: Normal driving ability (slow slew rate) 1: High driving ability (fast slew rate) Note: For the characteristics of the driving ability, refer to section 34.3.14, Output Slew Rate.
6	PB6DSR	0	R/W	
5	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
4	PB4DSR	0	R/W	When these bits are set to 1, the driving ability of the corresponding pin is higher than normal. 0: Normal driving ability (slow slew rate) 1: High driving ability (fast slew rate) Note: For the characteristics of the driving ability, refer to section 34.3.14, Output Slew Rate.
3	PB3DSR	0	R/W	
2	PB2DSR	0	R/W	
1	PB1DSR	0	R/W	
0	PB0DSR	0	R/W	

24.4.5 Port B Pin State Setting Register (PBPSR)

PBPSR is a 16-bit readable/writable register that enables or disables the pull-down resistor of port B. Bits PB8PSR, PB5PSR, PB3PSR, and PB1PSR correspond to pins PB8/$\overline{\text{WAIT}}$/TOE20/TIF20/RxD_A, PB5/A21/CRx_B/TIF7/TIF27/RxD_B, PB3/A19/MISOB/TIA05/SSLA7, and PB1/A17/MISOA/FRRxD_B, respectively. The PBPSR setting is ignored when the corresponding pin function is specified as a bus function ($\overline{\text{WAIT}}$, A21, A19 and A17).

Setting the specific bits of PBPSR to 1 puts their corresponding pins into pull-down state.

PBPSR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

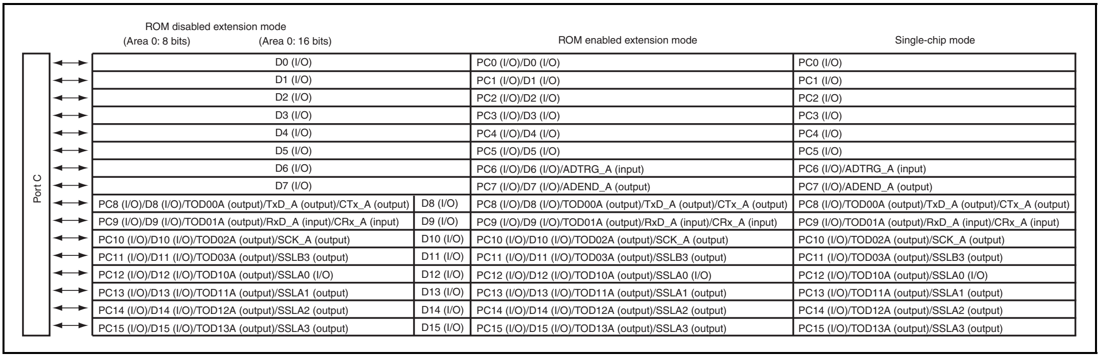
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	PB8 PSR	-	-	PB5 PSR	-	PB3 PSR	-	PB1 PSR	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R	R	R/W	R	R/W	R	R/W	R

Bit	Bit Name	Initial Value	R/W	Descriptions
15 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	PB8PSR	0	R/W	Selects whether or not the pull-down resistor of a corresponding pin is enabled. 0: Pull-down resistor disabled 1: Pull-down resistor enabled
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	PB5PSR	0	R/W	Selects whether or not the pull-down resistor of a corresponding pin is enabled. 0: Pull-down resistor disabled 1: Pull-down resistor enabled
4	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Descriptions
3	PB3PSR	0	R/W	Selects whether or not the pull-down resistor of a corresponding pin is enabled. 0: Pull-down resistor disabled 1: Pull-down resistor enabled
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1	PB1PSR	0	R/W	Selects whether or not the pull-down resistor of a corresponding pin is enabled. 0: Pull-down resistor disabled 1: Pull-down resistor enabled
0	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

24.5 Port C

Port C is an input/output port with the 16 pins shown in figure 24.3.



ROM disabled extension mode (Area 0: 8 bits)	ROM disabled extension mode (Area 0: 16 bits)	ROM enabled extension mode	Single-chip mode
D0 (I/O)		PC0 (I/O)/D0 (I/O)	PC0 (I/O)
D1 (I/O)		PC1 (I/O)/D1 (I/O)	PC1 (I/O)
D2 (I/O)		PC2 (I/O)/D2 (I/O)	PC2 (I/O)
D3 (I/O)		PC3 (I/O)/D3 (I/O)	PC3 (I/O)
D4 (I/O)		PC4 (I/O)/D4 (I/O)	PC4 (I/O)
D5 (I/O)		PC5 (I/O)/D5 (I/O)	PC5 (I/O)
D6 (I/O)		PC6 (I/O)/D6 (I/O)/ADTRG_A (input)	PC6 (I/O)/ADTRG_A (input)
D7 (I/O)		PC7 (I/O)/D7 (I/O)/ADEND_A (output)	PC7 (I/O)/ADEND_A (output)
PC8 (I/O)/D8 (I/O)/TOD00A (output)/TxD_A (output)/CTx_A (output)	D8 (I/O)	PC8 (I/O)/D8 (I/O)/TOD00A (output)/TxD_A (output)/CTx_A (output)	PC8 (I/O)/TOD00A (output)/TxD_A (output)/CTx_A (output)
PC9 (I/O)/D9 (I/O)/TOD01A (output)/RxD_A (input)/CRx_A (input)	D9 (I/O)	PC9 (I/O)/D9 (I/O)/TOD01A (output)/RxD_A (input)/CRx_A (input)	PC9 (I/O)/TOD01A (output)/RxD_A (input)/CRx_A (input)
PC10 (I/O)/D10 (I/O)/TOD02A (output)/SCK_A (output)	D10 (I/O)	PC10 (I/O)/D10 (I/O)/TOD02A (output)/SCK_A (output)	PC10 (I/O)/TOD02A (output)/SCK_A (output)
PC11 (I/O)/D11 (I/O)/TOD03A (output)/SSLB3 (output)	D11 (I/O)	PC11 (I/O)/D11 (I/O)/TOD03A (output)/SSLB3 (output)	PC11 (I/O)/TOD03A (output)/SSLB3 (output)
PC12 (I/O)/D12 (I/O)/TOD10A (output)/SSLA0 (I/O)	D12 (I/O)	PC12 (I/O)/D12 (I/O)/TOD10A (output)/SSLA0 (I/O)	PC12 (I/O)/TOD10A (output)/SSLA0 (I/O)
PC13 (I/O)/D13 (I/O)/TOD11A (output)/SSLA1 (output)	D13 (I/O)	PC13 (I/O)/D13 (I/O)/TOD11A (output)/SSLA1 (output)	PC13 (I/O)/TOD11A (output)/SSLA1 (output)
PC14 (I/O)/D14 (I/O)/TOD12A (output)/SSLA2 (output)	D14 (I/O)	PC14 (I/O)/D14 (I/O)/TOD12A (output)/SSLA2 (output)	PC14 (I/O)/TOD12A (output)/SSLA2 (output)
PC15 (I/O)/D15 (I/O)/TOD13A (output)/SSLA3 (output)	D15 (I/O)	PC15 (I/O)/D15 (I/O)/TOD13A (output)/SSLA3 (output)	PC15 (I/O)/TOD13A (output)/SSLA3 (output)

Figure 24.3 Port C

24.5.1 Port C Data Register (PCDR)

PCDR is a 16-bit readable/writable register that stores port C data. Bits PC15DR to PC0DR correspond to pins PC15/D15/TOD13A/SSLA3 to PC0/D0, respectively.

When a pin functions as a general output, a value written to a corresponding bit in this register is output on the pin. The bit value can be read directly regardless of the pin state by reading the bit in this register.

When this register is read while a pin functions as a general input, the pin state not a register value can be read. When data is written to this register while the pin functions as a general input, data can be written but it does not affect the pin state. Table 24.4 summarizes the PCDR read and write operations.

PCDR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PC15 DR	PC14 DR	PC13 DR	PC12 DR	PC11 DR	PC10 DR	PC9DR	PC8DR	PC7DR	PC6DR	PC5DR	PC4DR	PC3DR	PC2DR	PC1DR	PC0DR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Descriptions
15	PC15DR	0	R/W	Refer to table 24.4.
14	PC14DR	0	R/W	
13	PC13DR	0	R/W	
12	PC12DR	0	R/W	
11	PC11DR	0	R/W	
10	PC10DR	0	R/W	
9	PC9DR	0	R/W	
8	PC8DR	0	R/W	
7	PC7DR	0	R/W	
6	PC6DR	0	R/W	
5	PC5DR	0	R/W	
4	PC4DR	0	R/W	
3	PC3DR	0	R/W	
2	PC2DR	0	R/W	
1	PC1DR	0	R/W	
0	PC0DR	0	R/W	

Table 24.4 Read and Write Operations of Port C Data Register (PCDR)

- Bits 15 to 0 in PCDR

PCIOR	Pin Function	Read	Write
0	General input	Pin state	Data can be written to PCDR but data does not affect the pin state.
	Other than general input	Pin state	Data can be written to PCDR but data does not affect the pin state.
1	General output	PCDR value	Data written to PCDR is output on the pin.
	Other than general output	PCDR value	Data can be written to PCDR but data does not affect the pin state.

24.5.2 Port C Port Register (PCPR)

PCPR is a 16-bit read-only register that always stores port C pin states. This register cannot be directly written by the CPU. Bits PC15PR to PC0PR correspond to pins PC15/D15/TOD13A/SSLA3 to PC0/D0, respectively.

When PCPR is read, the pin state can be read.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PC15PR	PC14PR	PC13PR	PC12PR	PC11PR	PC10PR	PC9PR	PC8PR	PC7PR	PC6PR	PC5PR	PC4PR	PC3PR	PC2PR	PC1PR	PC0PR
Initial value:	PC15	PC14	PC13	PC12	PC11	PC10	PC9	PC8	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Descriptions
15	PC15PR	Pin state	R	When these bits are read, the pin states can be read. These bits cannot be modified.
14	PC14PR	Pin state	R	
13	PC13PR	Pin state	R	
12	PC12PR	Pin state	R	
11	PC11PR	Pin state	R	
10	PC10PR	Pin state	R	
9	PC9PR	Pin state	R	
8	PC8PR	Pin state	R	
7	PC7PR	Pin state	R	
6	PC6PR	Pin state	R	
5	PC5PR	Pin state	R	
4	PC4PR	Pin state	R	
3	PC3PR	Pin state	R	
2	PC2PR	Pin state	R	
1	PC1PR	Pin state	R	
0	PC0PR	Pin state	R	

24.5.3 Port C Inverting Register (PCIR)

PCIR is a 16-bit readable/writable register that enables or disables the inverting function for port C. Bits PC15IR to PC8IR correspond to pins PC15/D15/TOD13A/SSLA3 to PC8/D8/TOD00A/TxD_A/CTx_A, respectively. The PCIR setting is ignored when the corresponding pin function is specified as a bus function.

Setting the specific bits of PCIR to 1 reverts the output values on their corresponding pins.

PCIR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PC15IR	PC14IR	PC13IR	PC12IR	PC11IR	PC10IR	PC9IR	PC8IR	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Descriptions
15	PC15IR	0	R/W	These bits set whether or not the set value is inverted when it is output on a corresponding pin. 0: The set value is not inverted when output 1: The set value is inverted when output
14	PC14IR	0	R/W	
13	PC13IR	0	R/W	
12	PC12IR	0	R/W	
11	PC11IR	0	R/W	
10	PC10IR	0	R/W	
9	PC9IR	0	R/W	
8	PC8IR	0	R/W	
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

24.5.4 Port C Driving Ability Setting Register (PCDSR)

PCDSR is a 16-bit readable/writable register that selects the driving ability for port C. Bits PC15DSR to PC10DSR and PC8DSR correspond to pins PC15/D15/TOD13A/SSLA3 to PC10/D10/TOD02A/SCK_A and PC8/D8/TOD00A/TxD_A/CTx_A, respectively. The PCDSR setting is ignored when the corresponding pin function is specified as bus function (D15 to D10 and D8).

Setting the specific bits of PCDSR to 1 increases the driving ability of their corresponding pins.

PCDSR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PC15 DSR	PC14 DSR	PC13 DSR	PC12 DSR	PC11 DSR	PC10 DSR	-	PC8 DSR	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Descriptions
15	PC15DSR	0	R/W	When these bits are set to 1, the driving ability of the corresponding pin is higher than normal. 0: Normal driving ability (slow slew rate) 1: High driving ability (fast slew rate) Note: For the characteristics of the driving ability, refer to section 34.3.14, Output Slew Rate.
14	PC14DSR	0	R/W	
13	PC13DSR	0	R/W	
12	PC12DSR	0	R/W	
11	PC11DSR	0	R/W	
10	PC10DSR	0	R/W	
9	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
8	PC8DSR	0	R/W	When these bits are set to 1, the driving ability of the corresponding pin is higher than normal. 0: Normal driving ability (slow slew rate) 1: High driving ability (fast slew rate) Note: For the characteristics of the driving ability, refer to section 34.3.14, Output Slew Rate.
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

24.5.5 Port C Pin State Setting Register (PCPSR)

PCPSR is a 16-bit readable/writable register that enables or disables the pull-down resistor of port C. Bit PC9PSR corresponds to pin PC9/D9/TOD01A/RxD_A/CRx_A. The PCPSR setting is ignored when the corresponding pin function is specified as a bus function (D9).

Setting the specific bit of PCPSR to 1 puts its corresponding pin into pull-down state.

PCPSR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	PC9PSR	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Descriptions
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	PC9PSR	0	R/W	Selects whether or not the pull-down resistor of a corresponding pin is enabled. 0: Pull-down resistor disabled 1: Pull-down resistor enabled
8 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

24.6 Port D

Port D is an input/output port with the 14 pins shown in figure 24.4.

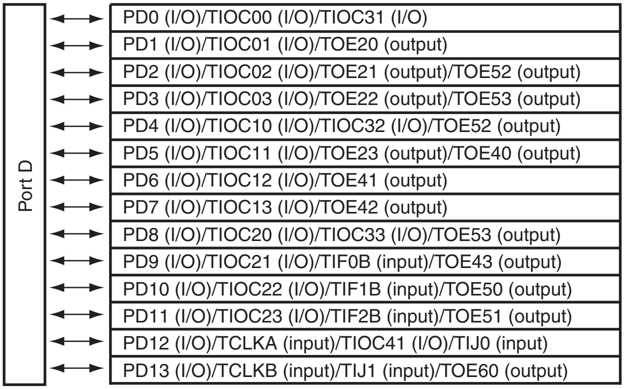

Port D
PD0 (I/O)/TIOC00 (I/O)/TIOC31 (I/O)
PD1 (I/O)/TIOC01 (I/O)/TOE20 (output)
PD2 (I/O)/TIOC02 (I/O)/TOE21 (output)/TOE52 (output)
PD3 (I/O)/TIOC03 (I/O)/TOE22 (output)/TOE53 (output)
PD4 (I/O)/TIOC10 (I/O)/TIOC32 (I/O)/TOE52 (output)
PD5 (I/O)/TIOC11 (I/O)/TOE23 (output)/TOE40 (output)
PD6 (I/O)/TIOC12 (I/O)/TOE41 (output)
PD7 (I/O)/TIOC13 (I/O)/TOE42 (output)
PD8 (I/O)/TIOC20 (I/O)/TIOC33 (I/O)/TOE53 (output)
PD9 (I/O)/TIOC21 (I/O)/TIF0B (input)/TOE43 (output)
PD10 (I/O)/TIOC22 (I/O)/TIF1B (input)/TOE50 (output)
PD11 (I/O)/TIOC23 (I/O)/TIF2B (input)/TOE51 (output)
PD12 (I/O)/TCLKA (input)/TIOC41 (I/O)/TIJ0 (input)
PD13 (I/O)/TCLKB (input)/TIJ1 (input)/TOE60 (output)

Figure 24.4 Port D

24.6.1 Port D Data Register (PDDR)

PDDR is a 16-bit readable/writable register that stores port D data. Bits PD13DR to PD0DR correspond to pins PD13/TCLKB/TIJ1/TOE60 to PD0/TIOC00/TIOC31, respectively.

When a pin functions as a general output, a value written to a corresponding bit in this register is output on the pin. The bit value can be read directly regardless of the pin state by reading the bit in this register.

When this register is read while a pin functions as a general input, the pin state not a register value can be read. When data is written to this register while the pin functions as a general input, data can be written but it does not affect the pin state. Table 24.5 summarizes the PDDR read and write operations.

PDDR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PD13 DR	PD12 DR	PD11 DR	PD10 DR	PD9DR	PD8DR	PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Descriptions
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	PD13DR	0	R/W	Refer to table 24.5.
12	PD12DR	0	R/W	
11	PD11DR	0	R/W	
10	PD10DR	0	R/W	
9	PD9DR	0	R/W	
8	PD8DR	0	R/W	
7	PD7DR	0	R/W	
6	PD6DR	0	R/W	
5	PD5DR	0	R/W	
4	PD4DR	0	R/W	
3	PD3DR	0	R/W	
2	PD2DR	0	R/W	
1	PD1DR	0	R/W	
0	PD0DR	0	R/W	

Table 24.5 Read and Write Operations of Port D Data Register (PDDR)

- Bits 13 to 0 in the PDDR register

PDIOR	Pin Function	Read	Write
0	General input	Pin state	Data can be written to PDDR but data does not affect the pin state.
	Other than general input	Pin state	Data can be written to PDDR but data does not affect the pin state.
1	General output	PDDR value	Data written to PDDR is output on the pin.
	Other than general output	PDDR value	Data can be written to PDDR but data does not affect the pin state.

24.6.2 Port D Port Register (PDPR)

PDPR is a 16-bit read-only register that always stores port D pin states. This register cannot be directly written by the CPU. The PD13PR to PB0PR bits correspond to the PD13/TCLKB/TIJ1/TOE60 to PD0/TIOC00/TIOC31 pins, respectively.

When PDPR is read, the pin state can be read.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PD13PR	PD12PR	PD11PR	PD10PR	PD9PR	PD8PR	PD7PR	PD6PR	PD5PR	PD4PR	PD3PR	PD2PR	PD1PR	PD0PR
Initial value:	0	0	PD13	PD12	PD11	PD10	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Descriptions
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	PD13PR	Pin state	R	When these bits are read, the pin states can be read. These bits cannot be modified.
12	PD12PR	Pin state	R	
11	PD11PR	Pin state	R	
10	PD10PR	Pin state	R	
9	PD9PR	Pin state	R	
8	PD8PR	Pin state	R	
7	PD7PR	Pin state	R	
6	PD6PR	Pin state	R	
5	PD5PR	Pin state	R	
4	PD4PR	Pin state	R	
3	PD3PR	Pin state	R	
2	PD2PR	Pin state	R	
1	PD1PR	Pin state	R	
0	PD0PR	Pin state	R	

24.6.3 Port D Inverting Register (PDIR)

PDIR is a 16-bit readable/writable register that enables or disables the inverting function for port D. Bits PD13IR to PD0IR correspond to pins PD13/TCLKB/TIJ1/TOE60 to PD0/TIOC00/TIOC31, respectively. The PDIR setting is valid regardless of the port D pin functions.

Setting the specific bits of PDIR to 1 reverts the output values on their corresponding pins.

PDIR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PD13IR	PD12IR	PD11IR	PD10IR	PD9IR	PD8IR	PD7IR	PD6IR	PD5IR	PD4IR	PD3IR	PD2IR	PD1IR	PD0IR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Descriptions
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	PD13IR	0	R/W	These bits set whether or not the set value is inverted when it is output on a corresponding pin. 0: The set value is not inverted when output 1: The set value is inverted when output
12	PD12IR	0	R/W	
11	PD11IR	0	R/W	
10	PD10IR	0	R/W	
9	PD9IR	0	R/W	
8	PD8IR	0	R/W	
7	PD7IR	0	R/W	
6	PD6IR	0	R/W	
5	PD5IR	0	R/W	
4	PD4IR	0	R/W	
3	PD3IR	0	R/W	
2	PD2IR	0	R/W	
1	PD1IR	0	R/W	
0	PD0IR	0	R/W	

24.7 Port E

Port E is an input/output port with the 14 pins shown in figure 24.5.

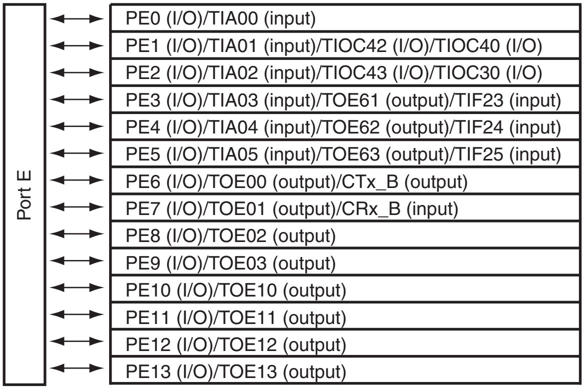

Port E
PE0 (I/O)/TIA00 (input)
PE1 (I/O)/TIA01 (input)/TIOC42 (I/O)/TIOC40 (I/O)
PE2 (I/O)/TIA02 (input)/TIOC43 (I/O)/TIOC30 (I/O)
PE3 (I/O)/TIA03 (input)/TOE61 (output)/TIF23 (input)
PE4 (I/O)/TIA04 (input)/TOE62 (output)/TIF24 (input)
PE5 (I/O)/TIA05 (input)/TOE63 (output)/TIF25 (input)
PE6 (I/O)/TOE00 (output)/CTx_B (output)
PE7 (I/O)/TOE01 (output)/CRx_B (input)
PE8 (I/O)/TOE02 (output)
PE9 (I/O)/TOE03 (output)
PE10 (I/O)/TOE10 (output)
PE11 (I/O)/TOE11 (output)
PE12 (I/O)/TOE12 (output)
PE13 (I/O)/TOE13 (output)

Figure 24.5 Port E

24.7.1 Port E Data Register (PEDR)

PEDR is a 16-bit readable/writable register that stores port E data. Bits PE13DR to PE0DR correspond to pins PE13/TOE13 to PE0/TIA00, respectively.

When a pin functions as a general output, a value written to a corresponding bit in this register is output on the pin. The bit value can be read directly regardless of the pin state by reading the bit in this register.

When this register is read while a pin functions as a general input, the pin state not a register value can be read. When data is written to this register while the pin functions as a general input, data can be written but it does not affect the pin state. Table 24.6 summarizes the PEDR read and write operations.

PEDR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PE13 DR	PE12 DR	PE11 DR	PE10 DR	PE9DR	PE8DR	PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR	PE0DR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Descriptions
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	PE13DR	0	R/W	Refer to table 24.6.
12	PE12DR	0	R/W	
11	PE11DR	0	R/W	
10	PE10DR	0	R/W	
9	PE9DR	0	R/W	
8	PE8DR	0	R/W	
7	PE7DR	0	R/W	
6	PE6DR	0	R/W	
5	PE5DR	0	R/W	
4	PE4DR	0	R/W	
3	PE3DR	0	R/W	
2	PE2DR	0	R/W	
1	PE1DR	0	R/W	
0	PE0DR	0	R/W	

Table 24.6 Read and Write Operations of Port E Data Register (PEDR)

- Bits 13 to 0 in the PEDR register

PEIOR	Pin Function	Read	Write
0	General input	Pin state	Data can be written to PEDR but data does not affect the pin state.
	Other than general input	Pin state	Data can be written to PEDR but data does not affect the pin state.
1	General output	PEDR value	Data written to PEDR is output on the pin.
	Other than general output	PEDR value	Data can be written to PEDR but data does not affect the pin state.

24.7.2 Port E Port Register (PEPR)

PEPR is a 16-bit read-only register that always stores port E pin states. This register cannot be directly written by the CPU. The PE13PR to PE0PR bits correspond to the PE13/TOE13 to PE0/TIA00 pins, respectively.

When PEPR is read, the pin state can be read.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PE13 PR	PE12 PR	PE11 PR	PE10 PR	PE9PR	PE8PR	PE7PR	PE6PR	PE5PR	PE4PR	PE3PR	PE2PR	PE1PR	PE0PR
Initial value:	0	0	PE13	PE12	PE11	PE10	PE9	PE8	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Descriptions
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	PE13PR	Pin state	R	When these bits are read, the pin states can be read. These bits cannot be modified.
12	PE12PR	Pin state	R	
11	PE11PR	Pin state	R	
10	PE10PR	Pin state	R	
9	PE9PR	Pin state	R	
8	PE8PR	Pin state	R	
7	PE7PR	Pin state	R	
6	PE6PR	Pin state	R	
5	PE5PR	Pin state	R	
4	PE4PR	Pin state	R	
3	PE3PR	Pin state	R	
2	PE2PR	Pin state	R	
1	PE1PR	Pin state	R	
0	PE0PR	Pin state	R	

24.7.3 Port E Inverting Register (PEIR)

PEIR is a 16-bit readable/writable register that enables or disables the inverting function for port E. Bits PE13IR to PE1IR correspond to pins PE13/TOE13 to PE1/TIA01/TIOC42/TIOC40, respectively. The PEIR setting is valid regardless of the port E pin functions.

Setting the specific bits of PEIR to 1 reverts the output values on their corresponding pins.

PEIR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PE13IR	PE12IR	PE11IR	PE10IR	PE9IR	PE8IR	PE7IR	PE6IR	PE5IR	PE4IR	PE3IR	PE2IR	PE1IR	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	Bit Name	Initial Value	R/W	Descriptions
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	PE13IR	0	R/W	These bits set whether or not the set value is inverted when it is output on a corresponding pin. 0: The set value is not inverted when output 1: The set value is inverted when output
12	PE12IR	0	R/W	
11	PE11IR	0	R/W	
10	PE10IR	0	R/W	
9	PE9IR	0	R/W	
8	PE8IR	0	R/W	
7	PE7IR	0	R/W	
6	PE6IR	0	R/W	
5	PE5IR	0	R/W	
4	PE4IR	0	R/W	
3	PE3IR	0	R/W	
2	PE2IR	0	R/W	
1	PE1IR	0	R/W	
0	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

24.7.4 Port E Driving Ability Setting Register (PEDSR)

PEDSR is a 16-bit readable/writable register that selects the driving ability for port E. Bits PE10DSR to PE6DSR correspond to pins PE10/TOE10 to PE6/TOE00/CTx_B, respectively. The PEDSR setting is valid regardless of the selected pin functions.

Setting the specific bits of PEDSR to 1 increases the driving ability of their corresponding pins.

PEDSR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	PE10 DSR	PE9 DSR	PE8 DSR	PE7 DSR	PE6 DSR	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Descriptions
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10	PE10DSR	0	R/W	When these bits are set to 1, the driving ability of the corresponding pin is higher than normal. 0: Normal driving ability (slow slew rate) 1: High driving ability (fast slew rate) Note: For the characteristics of the driving ability, refer to section 34.3.14, Output Slew Rate.
9	PE9DSR	0	R/W	
8	PE8DSR	0	R/W	
7	PE7DSR	0	R/W	
6	PE6DSR	0	R/W	
5 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

24.8 Port F

Port F is an input/output port with the 16 pins shown in figure 24.6.

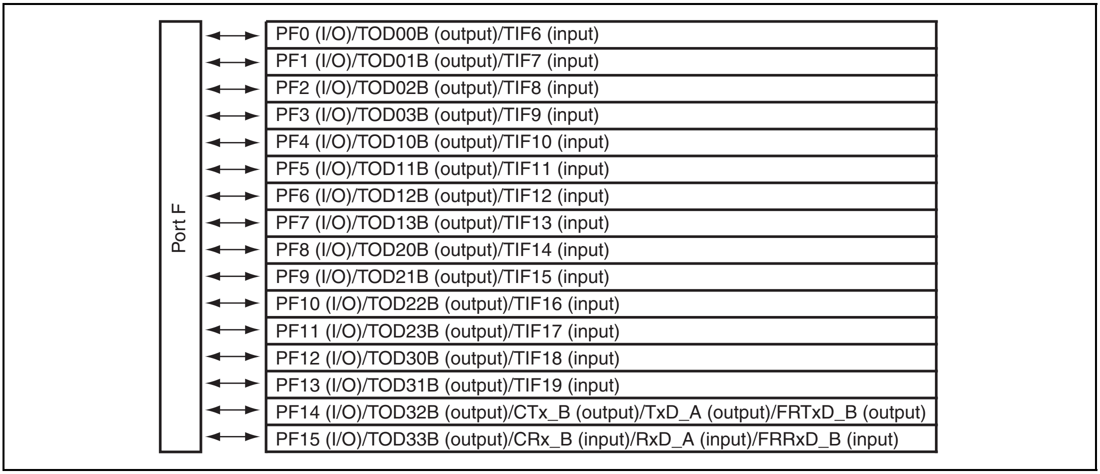

Port F
PF0 (I/O)/TOD00B (output)/TIF6 (input)
PF1 (I/O)/TOD01B (output)/TIF7 (input)
PF2 (I/O)/TOD02B (output)/TIF8 (input)
PF3 (I/O)/TOD03B (output)/TIF9 (input)
PF4 (I/O)/TOD10B (output)/TIF10 (input)
PF5 (I/O)/TOD11B (output)/TIF11 (input)
PF6 (I/O)/TOD12B (output)/TIF12 (input)
PF7 (I/O)/TOD13B (output)/TIF13 (input)
PF8 (I/O)/TOD20B (output)/TIF14 (input)
PF9 (I/O)/TOD21B (output)/TIF15 (input)
PF10 (I/O)/TOD22B (output)/TIF16 (input)
PF11 (I/O)/TOD23B (output)/TIF17 (input)
PF12 (I/O)/TOD30B (output)/TIF18 (input)
PF13 (I/O)/TOD31B (output)/TIF19 (input)
PF14 (I/O)/TOD32B (output)/CTx_B (output)/TxD_A (output)/FRTxD_B (output)
PF15 (I/O)/TOD33B (output)/CRx_B (input)/RxD_A (input)/FRRxD_B (input)

Figure 24.6 Port F

24.8.1 Port F Data Register (PFDR)

PFDR is a 16-bit readable/writable register that stores port F data. Bits PF15DR to PF0DR correspond to pins PF15/TOD33B/CRx_B/RxD_A/FRRxD_B to PF0/TOD00B/TIF6, respectively.

When a pin functions as a general output, a value written to a corresponding bit in this register is output on the pin. The bit value can be read directly regardless of the pin state by reading the bit in this register.

When this register is read while a pin functions as a general input, the pin state not a register value can be read. When data is written to this register while the pin functions as a general input, data can be written but it does not affect the pin state. Table 24.7 summarizes the PFDR read and write operations.

PFDR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PF15DR	PF14DR	PF13DR	PF12DR	PF11DR	PF10DR	PF9DR	PF8DR	PF7DR	PF6DR	PF5DR	PF4DR	PF3DR	PF2DR	PF1DR	PF0DR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Descriptions
15	PF15DR	0	R/W	Refer to table 24.7.
14	PF14DR	0	R/W	
13	PF13DR	0	R/W	
12	PF12DR	0	R/W	
11	PF11DR	0	R/W	
10	PF10DR	0	R/W	
9	PF9DR	0	R/W	
8	PF8DR	0	R/W	
7	PF7DR	0	R/W	
6	PF6DR	0	R/W	
5	PF5DR	0	R/W	
4	PF4DR	0	R/W	
3	PF3DR	0	R/W	
2	PF2DR	0	R/W	
1	PF1DR	0	R/W	
0	PF0DR	0	R/W	

Table 24.7 Read and Write Operations of Port F Data Register (PFDR)

- Bits 15 to 0 in the PFDR register

PFIOR	Pin Function	Read	Write
0	General input	Pin state	Data can be written to PFDR but data does not affect the pin state.
	Other than general input	Pin state	Data can be written to PFDR but data does not affect the pin state.
1	General output	PFDR value	Data written to PFDR is output on the pin.
	Other than general output	PFDR value	Data can be written to PFDR but data does not affect the pin state.

24.8.2 Port F Port Register (PFPR)

PFPR is a 16-bit read-only register that always stores port F pin states. This register cannot be directly written by the CPU. Bits PF15PR to PF0PR correspond to pins PF15/TOD33B/CRx_B/RxD_A/FRRxD_B to PF0/TOD00B/TIF6, respectively.

When PFPR is read, the pin state can be read.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PF15 PR	PF14 PR	PF13 PR	PF12 PR	PF11 PR	PF10 PR	PF9PR	PF8PR	PF7PR	PF6PR	PF5PR	PF4PR	PF3PR	PF2PR	PF1PR	PF0PR
Initial value:	PF15	PF14	PF13	PF12	PF11	PF10	PF9	PF8	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Descriptions
15	PF15PR	Pin state	R	When these bits are read, the pin states can be read. These bits cannot be modified.
14	PF14PR	Pin state	R	
13	PF13PR	Pin state	R	
12	PF12PR	Pin state	R	
11	PF11PR	Pin state	R	
10	PF10PR	Pin state	R	
9	PF9PR	Pin state	R	
8	PF8PR	Pin state	R	
7	PF7PR	Pin state	R	
6	PF6PR	Pin state	R	
5	PF5PR	Pin state	R	
4	PF4PR	Pin state	R	
3	PF3PR	Pin state	R	
2	PF2PR	Pin state	R	
1	PF1PR	Pin state	R	
0	PF0PR	Pin state	R	

24.8.3 Port F Inverting Register (PFIR)

PFIR is a 16-bit readable/writable register that enables or disables the inverting function for port F. Bits PF15IR to PF0IR correspond to pins PF15/TOD33B/CRx_B/RxD_A/FRRxD_B to PF0/TOD00B/TIF6, respectively. The PFIR setting is valid regardless of the port F pin functions.

Setting the specific bits of PFIR to 1 reverts the output values on their corresponding pins.

PFIR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PF15IR	PF14IR	PF13IR	PF12IR	PF11IR	PF10IR	PF9IR	PF8IR	PF7IR	PF6IR	PF5IR	PF4IR	PF3IR	PF2IR	PF1IR	PF0IR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Descriptions
15	PF15IR	0	R/W	These bits set whether or not the set value is inverted when it is output on a corresponding pin. 0: The set value is not inverted when output 1: The set value is inverted when output
14	PF14IR	0	R/W	
13	PF13IR	0	R/W	
12	PF12IR	0	R/W	
11	PF11IR	0	R/W	
10	PF10IR	0	R/W	
9	PF9IR	0	R/W	
8	PF8IR	0	R/W	
7	PF7IR	0	R/W	
6	PF6IR	0	R/W	
5	PF5IR	0	R/W	
4	PF4IR	0	R/W	
3	PF3IR	0	R/W	
2	PF2IR	0	R/W	
1	PF1IR	0	R/W	
0	PF0IR	0	R/W	

24.8.4 Port F Driving Ability Setting Register (PFDSR)

PFDSR is a 16-bit readable/writable register that selects the driving ability for port F. Bit PF14DSR corresponds to pin PF14/TOD32B/CTx_B/TxD_A/FRTxD_B. The PFDSR setting is valid regardless of the selected pin functions.

Setting the specific bits of PFDSR to 1 increases the driving ability of their corresponding pins.

PFDSR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PF14 DSR	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Descriptions
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14	PF14DSR	0	R/W	When these bits are set to 1, the driving ability of the corresponding pin is higher than normal. 0: Normal driving ability (slow slew rate) 1: High driving ability (fast slew rate) Note: For the characteristics of the driving ability, refer to section 34.3.14, Output Slew Rate.
13 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

24.8.5 Port F Pin State Setting Register (PFPSR)

PFPSR is a 16-bit readable/writable register that enables or disables the pull-down state of port F. Bit PF15PSR corresponds to pins PF15/TOD33B/CRx_B/RxD_A/FRRxD_B.

Setting the specific bit of PFPSR to 1 puts its corresponding pin into pull-down state.

PFPSR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PF15 PSR	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Descriptions
15	PF15PSR	0	R/W	Selects whether or not to enable the pull-down resistor of a corresponding pin is enabled. 0: Pull-down resistor disabled 1: Pull-down resistor enabled
14 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

24.9 Port G

Port G is an input/output port with the 16 pins shown in figure 24.7.

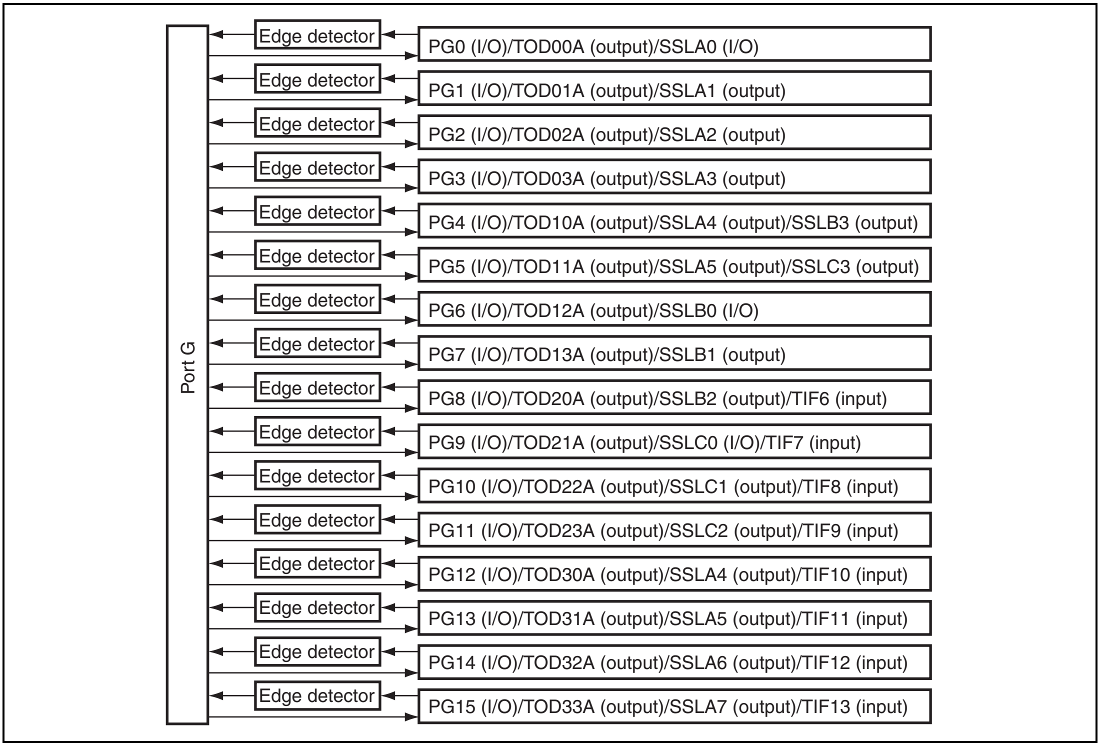

Port G
Edge detector
PG0 (I/O)/TOD00A (output)/SSLA0 (I/O)
PG1 (I/O)/TOD01A (output)/SSLA1 (output)
PG2 (I/O)/TOD02A (output)/SSLA2 (output)
PG3 (I/O)/TOD03A (output)/SSLA3 (output)
PG4 (I/O)/TOD10A (output)/SSLA4 (output)/SSLB3 (output)
PG5 (I/O)/TOD11A (output)/SSLA5 (output)/SSLC3 (output)
PG6 (I/O)/TOD12A (output)/SSLB0 (I/O)
PG7 (I/O)/TOD13A (output)/SSLB1 (output)
PG8 (I/O)/TOD20A (output)/SSLB2 (output)/TIF6 (input)
PG9 (I/O)/TOD21A (output)/SSLC0 (I/O)/TIF7 (input)
PG10 (I/O)/TOD22A (output)/SSLC1 (output)/TIF8 (input)
PG11 (I/O)/TOD23A (output)/SSLC2 (output)/TIF9 (input)
PG12 (I/O)/TOD30A (output)/SSLA4 (output)/TIF10 (input)
PG13 (I/O)/TOD31A (output)/SSLA5 (output)/TIF11 (input)
PG14 (I/O)/TOD32A (output)/SSLA6 (output)/TIF12 (input)
PG15 (I/O)/TOD33A (output)/SSLA7 (output)/TIF13 (input)

Figure 24.7 Port G

24.9.1 Port G Data Register (PGDR)

PGDR is a 16-bit readable/writable register that stores port G data. Bits PG15DR to PG0DR correspond to pins PG15/TOD33A/SSLA7/TIF13 to PG0/TOD00A/SSLA0, respectively.

When a pin functions as a general output, a value written to a corresponding bit in this register is output on the pin. The bit value can be read directly regardless of the pin state by reading the bit in this register.

When this register is read while a pin functions as a general input, the bit value can be read. Each bit of PGDR is set to 1 when an edge specified by the port G edge selecting register (PGER) is detected on the corresponding pin. Only 0 can be written after the bit is read as 1. Writing 0 without having read the bit as 1 or writing 1 is ignored. Writing 0 to the bits does not affect the pin state. Table 24.8 summarizes the PGDR read and write operations.

PGDR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PG15 DR	PG14 DR	PG13 DR	PG12 DR	PG11 DR	PG10 DR	PG9DR	PG8DR	PG7DR	PG6DR	PG5DR	PG4DR	PG3DR	PG2DR	PG1DR	PG0DR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Descriptions
15	PG15DR	0	R/W	Refer to table 24.8.
14	PG14DR	0	R/W	
13	PG13DR	0	R/W	
12	PG12DR	0	R/W	
11	PG11DR	0	R/W	
10	PG10DR	0	R/W	
9	PG9DR	0	R/W	
8	PG8DR	0	R/W	
7	PG7DR	0	R/W	
6	PG6DR	0	R/W	
5	PG5DR	0	R/W	
4	PG4DR	0	R/W	
3	PG3DR	0	R/W	
2	PG2DR	0	R/W	
1	PG1DR	0	R/W	
0	PG0DR	0	R/W	

Table 24.8 Read and Write Operations of Port G Data Register (PGDR)

- Bits 15 to 0 in the PGDR register

PGIOR	Pin Function	Read	Write
0	General input	PGDR value	Writing 1 is ignored. Only 0 can be written after this bit is read as 1. Writing 0 to this bit does not affect the pin state. Each bit can be set to 1 when an edge specified by PGER is input on the corresponding pin.
	Other than general input	PGDR value	Writing 1 is ignored. Only 0 can be written after this bit is read as 1. Writing 0 to this bit does not affect the pin state. Each bit can be set to 1 when an edge specified by PGER is input/output on the corresponding pin.
1	General output	PGDR value	Data written to PGDR is output on the pin.
	Other than general output	PGDR value	Data can be written to PGDR but data does not affect the pin state.

24.9.2 Port G Port Register (PGPR)

PGPR is a 16-bit read-only register that always stores port G pin states. This register cannot be directly written by the CPU. Bits PG15PR to PG0PR correspond to pins PG15/TOD33A/SSLA7/TIF13 to PG0/TOD00A/SSLA0, respectively.

When PGPR is read, the pin state can be read.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PG15PR	PG14PR	PG13PR	PG12PR	PG11PR	PG10PR	PG9PR	PG8PR	PG7PR	PG6PR	PG5PR	PG4PR	PG3PR	PG2PR	PG1PR	PG0PR
Initial value:	PG15	PG14	PG13	PG12	PG11	PG10	PG9	PG8	PG7	PG6	PG5	PG4	PG3	PG2	PG1	PG0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Descriptions
15	PG15PR	Pin state	R	When these bits are read, the pin states can be read. These bits cannot be modified.
14	PG14PR	Pin state	R	
13	PG13PR	Pin state	R	
12	PG12PR	Pin state	R	
11	PG11PR	Pin state	R	
10	PG10PR	Pin state	R	
9	PG9PR	Pin state	R	
8	PG8PR	Pin state	R	
7	PG7PR	Pin state	R	
6	PG6PR	Pin state	R	
5	PG5PR	Pin state	R	
4	PG4PR	Pin state	R	
3	PG3PR	Pin state	R	
2	PG2PR	Pin state	R	
1	PG1PR	Pin state	R	
0	PG0PR	Pin state	R	

24.9.3 Port G Inverting Register (PGIR)

PGIR is a 16-bit readable/writable register that enables or disables the inverting function for port G. Bits PG15IR to PG0IR correspond to pins PG15/TOD33A/SSLA7/TIF13 to PG0/TOD00A/SSLA0, respectively. The PGIR setting is valid regardless of the port G pin functions.

Setting a bit of PGIR to 1 reverts the output value on its corresponding pin.

The PGIR register is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PG15 IR	PG14 IR	PG13 IR	PG12 IR	PG11 IR	PG10 IR	PG9IR	PG8IR	PG7IR	PG6IR	PG5IR	PG4IR	PG3IR	PG2IR	PG1IR	PG0IR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Descriptions
15	PG15IR	0	R/W	These bits set whether or not the set value is inverted when it is output on a corresponding pin. 0: The set value is not inverted when output 1: The set value is inverted when output
14	PG14IR	0	R/W	
13	PG13IR	0	R/W	
12	PG12IR	0	R/W	
11	PG11IR	0	R/W	
10	PG10IR	0	R/W	
9	PG9IR	0	R/W	
8	PG8IR	0	R/W	
7	PG7IR	0	R/W	
6	PG6IR	0	R/W	
5	PG5IR	0	R/W	
4	PG4IR	0	R/W	
3	PG3IR	0	R/W	
2	PG2IR	0	R/W	
1	PG1IR	0	R/W	
0	PG0IR	0	R/W	

24.9.4 Port G Driving Ability Setting Register (PGDSR)

PGDSR is a 16-bit readable/writable register that selects the driving ability for port G. Bits PG15DSR to PG0DSR correspond to pins PG15/TOD33A/SSLA7/TIF13 to PG0/TOD00A/SSLA0, respectively. The PGDSR setting is valid regardless of the selected pin functions.

When a bit in PGDSR is set to 1, the high driving ability is set in the corresponding pin.

PGDSR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. PGDSR is not initialized in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PG15 DSR	PG14 DSR	PG13 DSR	PG12 DSR	PG11 DSR	PG10 DSR	PG9 DSR	PG8 DSR	PG7 DSR	PG6 DSR	PG5 DSR	PG4 DSR	PG3 DSR	PG2 DSR	PG1 DSR	PG0 DSR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Descriptions
15	PG15DSR	0	R/W	When these bits are set to 1, the driving ability of the corresponding pin is higher than normal. 0: Normal driving ability (slow slew rate) 1: High driving ability (fast slew rate) Note: For the characteristics of the driving ability, refer to section 34.3.14, Output Slew Rate.
14	PG14DSR	0	R/W	
13	PG13DSR	0	R/W	
12	PG12DSR	0	R/W	
11	PG11DSR	0	R/W	
10	PG10DSR	0	R/W	
9	PG9DSR	0	R/W	
8	PG8DSR	0	R/W	
7	PG7DSR	0	R/W	
6	PG6DSR	0	R/W	
5	PG5DSR	0	R/W	
4	PG4DSR	0	R/W	
3	PG3DSR	0	R/W	
2	PG2DSR	0	R/W	
1	PG1DSR	0	R/W	
0	PG0DSR	0	R/W	

24.9.5 Port G Edge Selecting Register (PGER)

PGER is a 16-bit readable/writable register that selects the type of an edge to be detected by port G. The PGHES[1:0] bits correspond to pins PG15/TOD33A/SSLA7/TIF13 to PG8/TOD20A/SSLB2/TIF6. The PGLES[1:0] bits correspond to pins PG7/TOD13A/SSLB1 to PG0/TOD00A/SSLA0 pins.

When a pin functions as a general input, each bit in PGDR functions as a status flag indicating whether or not the specified edge input is detected.

PGER is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	PGHES[1:0]		-	-	-	-	-	-	PGLES[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Descriptions
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	PGHES[1:0]	00	R/W	These bits set the input edge to be detected on the corresponding pins PG15/TOD33A/SSLA7/TIF13 to PG8/TOD20A/SSLB2/TIF6. 00: No edge detected 01: Rising edge detected 10: Falling edge detected 11: Both rising and falling edges detected
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Descriptions
1, 0	PGLES[1:0]	00	R/W	These bits set the input edge to be detected on the corresponding pins PG7/TOD13A/SSLB1 to PG0/TOD00A/SSLA0. 00: No edge detected 01: Rising edge detected 10: Falling edge detected 11: Both rising and falling edges detected

24.10 Port H

Port H is an input/output port with the 6 pins shown in figure 24.8.

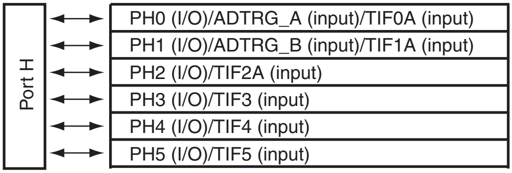


Figure 24.8 Port H

24.10.1 Port H Data Register (PHDR)

PHDR is a 16-bit readable/writable register that stores port H data. Bits PH5DR to PH0DR correspond to pins PH5/TIF5 to PH0/ADTRG_A/TIF0A, respectively.

When a pin functions as a general output, a value written to a corresponding bit in this register is output on the pin. The bit value can be read directly regardless of the pin state by reading the bit in this register.

When this register is read while a pin functions as a general input, the pin state not a register value can be read. When data is written to this register while the pin functions as a general input, data can be written but it does not affect the pin state. Table 24.9 summarizes the PHDR read and write operations.

PHDR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	PH5DR	PH4DR	PH3DR	PH2DR	PH1DR	PH0DR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Descriptions
15 to 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	PH5DR	0	R/W	Refer to table 24.9.
4	PH4DR	0	R/W	
3	PH3DR	0	R/W	
2	PH2DR	0	R/W	
1	PH1DR	0	R/W	
0	PH0DR	0	R/W	

Table 24.9 Read and Write Operations of Port H Data Register (PHDR)

- Bits 5 to 0 in the PHDR register

PHIOR	Pin Function	Read	Write
0	General input	Pin state	Data can be written to PHDR but data does not affect the pin state.
	Other than general input	Pin state	Data can be written to PHDR but data does not affect the pin state.
1	General output	PHDR value	Data written to PHDR is output on the pin.
	Other than general output	PHDR value	Data can be written to PHDR but data does not affect the pin state.

24.10.2 Port H Port Register (PHPR)

PHPR is a 16-bit read-only register that always stores port H pin states. This register cannot be directly written by the CPU. Bits PH5PR to PH0PR correspond to pins PH5/TIF5 to PH0/ADTRG_A/TIF0A, respectively.

When PHPR is read, the pin state can be read.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	PH5PR	PH4PR	PH3PR	PH2PR	PH1PR	PH0PR
Initial value:	0	0	0	0	0	0	0	0	0	0	PH5	PH4	PH3	PH2	PH1	PH0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Descriptions
15 to 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	PH5PR	Pin state	R	When these bits are read, the pin states can be read. These bits cannot be modified.
4	PH4PR	Pin state	R	
3	PH3PR	Pin state	R	
2	PH2PR	Pin state	R	
1	PH1PR	Pin state	R	
0	PH0PR	Pin state	R	

24.11 Port J

Port J is an input/output port with the 10 pins shown in figure 24.9.

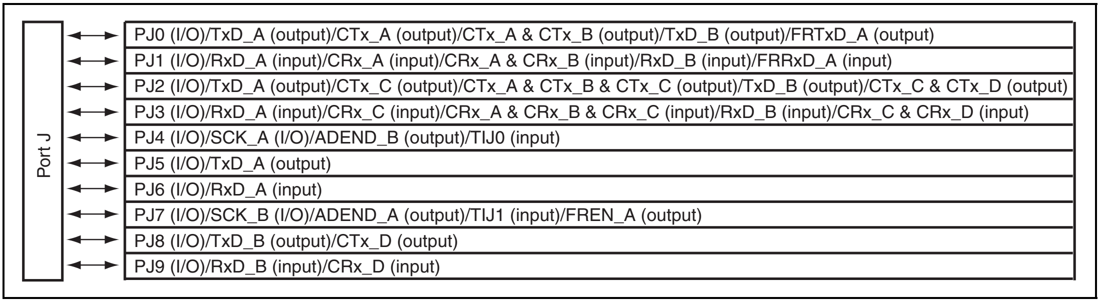

Port J
PJ0 (I/O)/TxD_A (output)/CTx_A (output)/CTx_A & CTx_B (output)/TxD_B (output)/FRTxD_A (output)
PJ1 (I/O)/RxD_A (input)/CRx_A (input)/CRx_A & CRx_B (input)/RxD_B (input)/FRRxD_A (input)
PJ2 (I/O)/TxD_A (output)/CTx_C (output)/CTx_A & CTx_B & CTx_C (output)/TxD_B (output)/CTx_C & CTx_D (output)
PJ3 (I/O)/RxD_A (input)/CRx_C (input)/CRx_A & CRx_B & CRx_C (input)/RxD_B (input)/CRx_C & CRx_D (input)
PJ4 (I/O)/SCK_A (I/O)/ADEND_B (output)/TIJ0 (input)
PJ5 (I/O)/TxD_A (output)
PJ6 (I/O)/RxD_A (input)
PJ7 (I/O)/SCK_B (I/O)/ADEND_A (output)/TIJ1 (input)/FREN_A (output)
PJ8 (I/O)/TxD_B (output)/CTx_D (output)
PJ9 (I/O)/RxD_B (input)/CRx_D (input)

Figure 24.9 Port J

24.11.1 Port J Data Register (PJDR)

PJDR is a 16-bit readable/writable register that stores port J data. Bits PJ9DR to PJ0DR correspond to pins PJ9/RxD_B/CRx_D to PJ0/TxD_A/CTx_A/CTx_A&CTx_B/TxD_B/FRTxD_A, respectively.

When a pin functions as a general output, a value written to a corresponding bit in this register is output on the pin. The bit value can be read directly regardless of the pin state by reading the bit in this register.

When this register is read while a pin functions as a general input, the pin state not a register value can be read. When data is written to this register while the pin functions as a general input, data can be written but it does not affect the pin state. Table 24.10 summarizes the PJDR read and write operations.

PJDR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	PJ9DR	PJ8DR	PJ7DR	PJ6DR	PJ5DR	PJ4DR	PJ3DR	PJ2DR	PJ1DR	PJ0DR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Descriptions
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	PJ9DR	0	R/W	Refer to table 24.10.
8	PJ8DR	0	R/W	
7	PJ7DR	0	R/W	
6	PJ6DR	0	R/W	
5	PJ5DR	0	R/W	
4	PJ4DR	0	R/W	
3	PJ3DR	0	R/W	
2	PJ2DR	0	R/W	
1	PJ1DR	0	R/W	
0	PJ0DR	0	R/W	

Table 24.10 Read and Write Operations of Port J Data Register (PJDR)

- Bits 9 to 0 in the PJDR register

PJIOR	Pin Function	Read	Write
0	General input	Pin state	Data can be written to PJDR but data does not affect the pin state.
	Other than general input	Pin state	Data can be written to PJDR but data does not affect the pin state.
1	General output	PJDR value	Data written to PJDR is output on the pin.
	Other than general output	PJDR value	Data can be written to PJDR but data does not affect the pin state.

24.11.2 Port J Port Register (PJPR)

PJPR is a 16-bit read-only register that always stores port J pin states. This register cannot be directly written by the CPU. Bits PJ9PR to PJ0PR correspond to pins PJ9/RxD_B/CRx_D to PJ0/TxD_A/CTx_A/CTx_A&CTx_B/TxD_B/FRTxD_A, respectively.

When PJPR is read, the pin state can be read.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	PJ9PR	PJ8PR	PJ7PR	PJ6PR	PJ5PR	PJ4PR	PJ3PR	PJ2PR	PJ1PR	PJ0PR
Initial value:	0	0	0	0	0	0	PJ9	PJ8	PJ7	PJ6	PJ5	PJ4	PJ3	PJ2	PJ1	PJ0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Descriptions
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	PJ9PR	Pin state	R	When these bits are read, the pin states can be read. These bits cannot be modified.
8	PJ8PR	Pin state	R	
7	PJ7PR	Pin state	R	
6	PJ6PR	Pin state	R	
5	PJ5PR	Pin state	R	
4	PJ4PR	Pin state	R	
3	PJ3PR	Pin state	R	
2	PJ2PR	Pin state	R	
1	PJ1PR	Pin state	R	
0	PJ0PR	Pin state	R	

24.11.3 Port J Inverting Register (PJIR)

PJIR is a 16-bit readable/writable register that enables or disables the inverting function for port J. Bits PJ8IR, PJ7IR, PJ5IR, PJ4IR, PJ2IR, and PJ0IR correspond to pins PJ8/TxD_B/CTx_D, PJ7/SCK_B/ADEND_A/TIJ1/FREN_A, PJ5/TxD_A, PJ4/SCK_A/ADEND_B/TIJ0, PJ2/TxD_A/CTx_C/CTx_A&CTx_B&CTx_C/TxD_B/CTx_C&CTx_D, and PJ0/TxD_A/CTx_A/CTx_A&CTx_B/TxD_B/FRTxD_A, respectively. The PJIR setting is valid regardless of the port J pin functions.

Setting the specific bits of PJIR to 1 inverts the output value on their corresponding pins.

PJIR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	PJ8IR	PJ7IR	-	PJ5IR	PJ4IR	-	PJ2IR	-	PJ0IR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W	R	R/W	R	R/W

Bit	Bit Name	Initial Value	R/W	Descriptions
15 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	PJ8IR	0	R/W	These bits set whether or not the set value is inverted when it is output on a corresponding pin. 0: The set value is not inverted when output 1: The set value is inverted when output
7	PJ7IR	0	R/W	
6	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
5	PJ5IR	0	R/W	These bits set whether or not the set value is inverted when it is output on a corresponding pin. 0: The set value is not inverted when output 1: The set value is inverted when output
4	PJ4IR	0	R/W	
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Descriptions
2	PJ2IR	0	R/W	Sets whether or not the set value is inverted when it is output on a corresponding pin. 0: The set value is not inverted when output 1: The set value is inverted when output
1	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
0	PJ0IR	0	R/W	Sets whether or not the set value is inverted when it is output on a corresponding pin. 0: The set value is not inverted when output 1: The set value is inverted when output

24.11.4 Port J Driving Ability Setting Register (PJDSR)

PJDSR is a 16-bit readable/writable register that selects the driving ability for port J. Bits PJ8DSR, PJ7DSR, PJ5DSR, PJ4DSR, PJ2DSR, and PJ0DSR correspond to pins PJ8/TxD_B/CTx_D, PJ7/SCK_B/ADEND_A/TIJ1/FREN_A, PJ5/TxD_A, PJ4/SCK_A/ADEND_B/TIJ0, PJ2/TxD_A/CTx_C/CTx_A&CTx_B&CTx_C/TxD_B/CTx_C&CTx_D, and PJ0/TxD_A/CTx_A/CTx_A&CTx_B/TxD_B/FRTxD_A, respectively. The PJDSR setting is always valid regardless of the selected pin functions.

Setting the specific bits of PJDSR to 1 increases the driving ability of their corresponding pins.

PJDSR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. PJDSR is not initialized in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	PJ8 DSR	PJ7 DSR	-	PJ5 DSR	PJ4 DSR	-	PJ2 DSR	-	PJ0 DSR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W	R	R/W	R	R/W

Bit	Bit Name	Initial Value	R/W	Descriptions
15 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	PJ8DSR	0	R/W	When these bits are set to 1, the driving ability of the corresponding pin is higher than normal. 0: Normal driving ability (slow slew rate) 1: High driving ability (fast slew rate) Note: For the characteristics of the driving ability, refer to section 34.3.14, Output Slew Rate.
7	PJ7DSR	0	R/W	
6	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
5	PJ5DSR	0	R/W	When these bits are set to 1, the driving ability of the corresponding pin is higher than normal. 0: Normal driving ability (slow slew rate) 1: High driving ability (fast slew rate) Note: For the characteristics of the driving ability, refer to section 34.3.14, Output Slew Rate.
4	PJ4DSR	0	R/W	
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	PJ2DSR	0	R/W	When these bits are set to 1, the driving ability of the corresponding pin is higher than normal. 0: Normal driving ability (slow slew rate) 1: High driving ability (fast slew rate) Note: For the characteristics of the driving ability, refer to section 34.3.14, Output Slew Rate.
1	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
0	PJ0DSR	0	R/W	When these bits are set to 1, the driving ability of the corresponding pin is higher than normal. 0: Normal driving ability (slow slew rate) 1: High driving ability (fast slew rate) Note: For the characteristics of the driving ability, refer to section 34.3.14, Output Slew Rate.

24.11.5 Port J Pin State Setting Register (PJPSR)

PJPSR is a 16-bit readable/writable register that enables or disables the pull-down resistor of port J. Bits PJ9PSR, PJ6PSR, PJ3PSR, and PJ1PSR correspond to pins PJ9/RxD_B/CRx_D, PJ6/RxD_A, PJ3/RxD_A/CRx_C/CRx_A&CRx_B&CRx_C/RxD_B/CRx_C&CRx_D, and PJ1/RxD_A/CRx_A/CRx_A&CRx_B/RxD_B/FRRxD_A, respectively. The PJPSR setting is valid regardless of the selected pin functions.

Setting the specific bits of PJPSR to 1 puts their corresponding pins into pull-down state.

PJPSR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	PJ9 PSR	-	-	PJ6 PSR	-	-	PJ3 PSR	-	PJ1 PSR	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R	R	R/W	R	R	R/W	R	R/W	R

Bit	Bit Name	Initial Value	R/W	Descriptions
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	PJ9PSR	0	R/W	Selects whether or not the pull-down resistor of a corresponding pin is enabled. 0: Pull-down resistor disabled 1: Pull-down resistor enabled
8, 7	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
6	PJ6PSR	0	R/W	Selects whether or not the pull-down resistor of a corresponding pin is enabled. 0: Pull-down resistor disabled 1: Pull-down resistor enabled
5, 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Descriptions
3	PJ3PSR	0	R/W	Selects whether or not the pull-down resistor of a corresponding pin is enabled. 0: Pull-down resistor disabled 1: Pull-down resistor enabled
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1	PJ1PSR	0	R/W	Selects whether or not the pull-down resistor of a corresponding pin is enabled. 0: Pull-down resistor disabled 1: Pull-down resistor enabled
0	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

24.12 Port K

Port K is an input/output port with the 12 pins shown in figure 24.10.

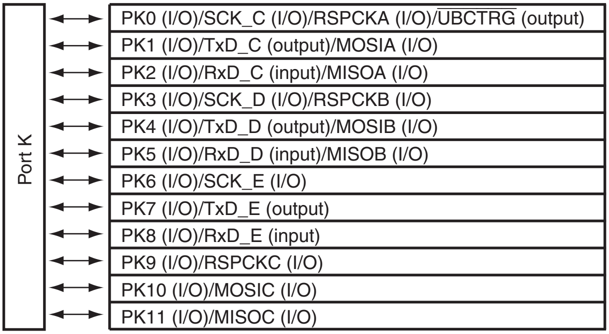

Port K
PK0 (I/O)/SCK_C (I/O)/RSPCKA (I/O)/$\overline{\text{UBCTRG}}$ (output)
PK1 (I/O)/TxD_C (output)/MOSIA (I/O)
PK2 (I/O)/RxD_C (input)/MISOA (I/O)
PK3 (I/O)/SCK_D (I/O)/RSPCKB (I/O)
PK4 (I/O)/TxD_D (output)/MOSIB (I/O)
PK5 (I/O)/RxD_D (input)/MISOB (I/O)
PK6 (I/O)/SCK_E (I/O)
PK7 (I/O)/TxD_E (output)
PK8 (I/O)/RxD_E (input)
PK9 (I/O)/RSPCKC (I/O)
PK10 (I/O)/MOSIC (I/O)
PK11 (I/O)/MISOC (I/O)

Figure 24.10 Port K

24.12.1 Port K Data Register (PKDR)

PKDR is a 16-bit readable/writable register that stores port K data. Bits PK11DR to PK0DR correspond to pins PK11/MISOC to PK0/SCK_C/RSPCKA/$\overline{\text{UBCTRG}}$, respectively.

When a pin functions as a general output, a value written to a corresponding bit in this register is output on the pin. The bit value can be read directly regardless of the pin state by reading the bit in this register.

When this register is read while a pin functions as a general input, the pin state not a register value can be read. When data is written to this register while the pin functions as a general input, data can be written but it does not affect the pin state. Table 24.11 summarizes the PKDR read and write operations.

PKDR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	PK11 DR	PK10 DR	PK9DR	PK8DR	PK7DR	PK6DR	PK5DR	PK4DR	PK3DR	PK2DR	PK1DR	PK0DR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Descriptions
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11	PK11DR	0	R/W	Refer to table 24.11.
10	PK10DR	0	R/W	
9	PK9DR	0	R/W	
8	PK8DR	0	R/W	
7	PK7DR	0	R/W	
6	PK6DR	0	R/W	
5	PK5DR	0	R/W	
4	PK4DR	0	R/W	
3	PK3DR	0	R/W	
2	PK2DR	0	R/W	
1	PK1DR	0	R/W	
0	PK0DR	0	R/W	

Table 24.11 Read and Write Operations of Port K Data Register (PKDR)

- Bits 11 to 0 in the PKDR register

PKIOR	Pin Function	Read	Write
0	General input	Pin state	Data can be written to PKDR but data does not affect the pin state.
	Other than general input	Pin state	Data can be written to PKDR but data does not affect the pin state.
1	General output	PKDR value	Data written to PKDR is output on the pin.
	Other than general output	PKDR value	Data can be written to PKDR but data does not affect the pin state.

24.12.2 Port K Port Register (PKPR)

PKPR is a 16-bit read-only register that always stores port K pin states. This register cannot be directly written by the CPU. The PK11PR to PK0PR bits correspond to the PK11/MISOC to PK0/SCK_C/RSPCKA/$\overline{\text{UBCTRG}}$ pins, respectively.

When PKPR is read, the pin state can be read.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	PK11PR	PK10PR	PK9PR	PK8PR	PK7PR	PK6PR	PK5PR	PK4PR	PK3PR	PK2PR	PK1PR	PK0PR
Initial value:	0	0	0	0	PK11	PK10	PK9	PK8	PK7	PK6	PK5	PK4	PK3	PK2	PK1	PK0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Descriptions
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11	PK11PR	Pin state	R	When these bits are read, the pin states can be read. These bits cannot be modified.
10	PK10PR	Pin state	R	
9	PK9PR	Pin state	R	
8	PK8PR	Pin state	R	
7	PK7PR	Pin state	R	
6	PK6PR	Pin state	R	
5	PK5PR	Pin state	R	
4	PK4PR	Pin state	R	
3	PK3PR	Pin state	R	
2	PK2PR	Pin state	R	
1	PK1PR	Pin state	R	
0	PK0PR	Pin state	R	

24.12.3 Port K Inverting Register (PKIR)

PKIR is a 16-bit readable/writable register that enables or disables the inverting function for port K. Bits PK11IR to PK9IR and PK7IR to PK0IR correspond to pins PK11/MISOC to PK9/RSPCKC and PK7/TxD_E to PK0/SCK_C/RSPCKA/$\overline{\text{UBCTRG}}$, respectively. The PKIR setting is valid regardless of the port K pin functions.

Setting the specific bits of PKIR to 1 inverts the output value on their corresponding pins.

PKIR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. PKIR is not initialized in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	PK11IR	PK10IR	PK9IR	-	PK7IR	PK6IR	PK5IR	PK4IR	PK3IR	PK2IR	PK1IR	PK0IR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Descriptions
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11	PK11IR	0	R/W	These bits set whether or not the set value is inverted when it is output on a corresponding pin. 0: The set value is not inverted when output 1: The set value is inverted when output
10	PK10IR	0	R/W	
9	PK9IR	0	R/W	
8	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
7	PK7IR	0	R/W	These bits set whether or not the set value is inverted when it is output on a corresponding pin. 0: The set value is not inverted when output 1: The set value is inverted when output
6	PK6IR	0	R/W	
5	PK5IR	0	R/W	
4	PK4IR	0	R/W	
3	PK3IR	0	R/W	
2	PK2IR	0	R/W	
1	PK1IR	0	R/W	
0	PK0IR	0	R/W	

24.12.4 Port K Driving Ability Setting Register (PKDSR)

PKDSR is a 16-bit readable/writable register that selects the driving ability for port K. Bits PK11DSR to PK9DSR and PK7DSR to PK0DSR correspond to pins PK11/MISOC to PK9/RSPCKC and PK7/TxD_E to PK0/SCK_C/RSPCKA/$\overline{\text{UBCTRG}}$, respectively. The PKDSR setting is valid regardless of the selected pin functions.

Setting the specific bits of PKDSR to 1 increases the driving ability of their corresponding pins.

PKDSR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	PK11 DSR	PK10 DSR	PK9 DSR	-	PK7 DSR	PK6 DSR	PK5 DSR	PK4 DSR	PK3 DSR	PK2 DSR	PK1 DSR	PK0 DSR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Descriptions
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11	PK11DSR	0	R/W	When these bits are set to 1, the driving ability of the corresponding pin is higher than normal. 0: Normal driving ability (slow slew rate) 1: High driving ability (fast slew rate) Note: For the characteristics of the driving ability, refer to section 34.3.14, Output Slew Rate.
10	PK10DSR	0	R/W	
9	PK9DSR	0	R/W	
8	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
7	PK7DSR	0	R/W	When these bits are set to 1, the driving ability of the corresponding pin is higher than normal. 0: Normal driving ability (slow slew rate) 1: High driving ability (fast slew rate) Note: For the characteristics of the driving ability, refer to section 34.3.14, Output Slew Rate.
6	PK6DSR	0	R/W	
5	PK5DSR	0	R/W	
4	PK4DSR	0	R/W	
3	PK3DSR	0	R/W	
2	PK2DSR	0	R/W	
1	PK1DSR	0	R/W	
0	PK0DSR	0	R/W	

24.12.5 Port K Pin State Setting Register (PKPSR)

PKPSR is a 16-bit readable/writable register that enables or disables the pull-down resistor of port K. Bits PK11PSR, PK8PSR, PK5PSR, and PK2PSR correspond to pins PK11/MISOC, PK8/RxD_E, PK5/RxD_D/MISOB, and PK2/RxD_C/MISOA, respectively. The PKPSR setting is valid regardless of the selected pin functions.

Setting the specific bits of PKPSR to 1 puts their corresponding pins into pull-down state.

PKPSR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	PK11 PSR	-	-	PK8 PSR	-	-	PK5 PSR	-	-	PK2 PSR	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R	R	R/W	R	R	R/W	R	R	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Descriptions
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11	PK11PSR	0	R/W	Selects whether or not the pull-down resistor of a corresponding pin is enabled. 0: Pull-down resistor disabled 1: Pull-down resistor enabled
10, 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	PK8PSR	0	R/W	Selects whether or not the pull-down resistor of a corresponding pin is enabled. 0: Pull-down resistor disabled 1: Pull-down resistor enabled
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Descriptions
5	PK5PSR	0	R/W	Selects whether or not the pull-down resistor of a corresponding pin is enabled. 0: Pull-down resistor disabled 1: Pull-down resistor enabled
4, 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	PK2PSR	0	R/W	Selects whether or not the pull-down resistor of a corresponding pin is enabled. 0: Pull-down resistor disabled 1: Pull-down resistor enabled
1, 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

24.13 Port L

Port L is an input/output port with the 9 pins shown in figure 24.11.

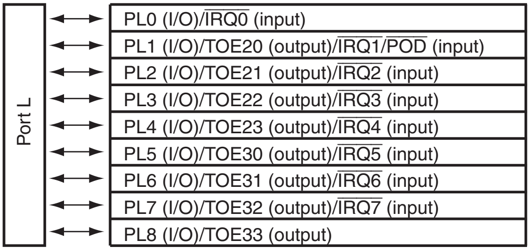

Port L
PL0 (I/O)/$\overline{\text{IRQ0}}$ (input)
PL1 (I/O)/TOE20 (output)/$\overline{\text{IRQ1}}$/$\overline{\text{POD}}$ (input)
PL2 (I/O)/TOE21 (output)/$\overline{\text{IRQ2}}$ (input)
PL3 (I/O)/TOE22 (output)/$\overline{\text{IRQ3}}$ (input)
PL4 (I/O)/TOE23 (output)/$\overline{\text{IRQ4}}$ (input)
PL5 (I/O)/TOE30 (output)/$\overline{\text{IRQ5}}$ (input)
PL6 (I/O)/TOE31 (output)/$\overline{\text{IRQ6}}$ (input)
PL7 (I/O)/TOE32 (output)/$\overline{\text{IRQ7}}$ (input)
PL8 (I/O)/TOE33 (output)

Figure 24.11 Port L

24.13.1 Port L Data Register (PLDR)

PLDR is a 16-bit readable/writable register that stores port L data. Bits PL8DR to PL0DR correspond to pins PL8/TOE33 to PL0/$\overline{\text{IRQ0}}$, respectively.

When a pin functions as a general output, a value written to a corresponding bit in this register is output on the pin. The bit value can be read directly regardless of the pin state by reading the bit in this register.

When this register is read while a pin functions as a general input, the pin state not a register value can be read. When data is written to this register while the pin functions as a general input, data can be written but it does not affect the pin state. Table 24.12 summarizes the PLDR read and write operations.

PLDR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	PL8DR	PL7DR	PL6DR	PL5DR	PL4DR	PL3DR	PL2DR	PL1DR	PL0DR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Descriptions
15 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	PL8DR	0	R/W	Refer to table 24.12.
7	PL7DR	0	R/W	
6	PL6DR	0	R/W	
5	PL5DR	0	R/W	
4	PL4DR	0	R/W	
3	PL3DR	0	R/W	
2	PL2DR	0	R/W	
1	PL1DR	0	R/W	
0	PL0DR	0	R/W	

Table 24.12 Read and Write Operations of Port L Data Register (PLDR)

- Bits 8 to 0 in the PLDR register

PLIOR	Pin Function	Read	Write
0	General input	Pin state	Data can be written to PLDR but data does not affect the pin state.
	Other than general input	Pin state	Data can be written to PLDR but data does not affect the pin state.
1	General output	PLDR value	Data written to PLDR is output on the pin.
	Other than general output	PLDR value	Data can be written to PLDR but data does not affect the pin state.

24.13.2 Port L Port Register (PLPR)

PLPR is a 16-bit read-only register that always stores port L pin states. This register cannot be directly written by the CPU. Bits PL8PR to PL0PR correspond to pins PL8/TOE33 to PL0/$\overline{\text{IRQ0}}$, respectively.

When PLPR is read, the pin state can be read.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	PL8PR	PL7PR	PL6PR	PL5PR	PL4PR	PL3PR	PL2PR	PL1PR	PL0PR
Initial value:	0	0	0	0	0	0	0	PL8	PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Descriptions
15 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	PL8PR	Pin state	R	When these bits are read, the pin states can be read. These bits cannot be modified.
7	PL7PR	Pin state	R	
6	PL6PR	Pin state	R	
5	PL5PR	Pin state	R	
4	PL4PR	Pin state	R	
3	PL3PR	Pin state	R	
2	PL2PR	Pin state	R	
1	PL1PR	Pin state	R	
0	PL0PR	Pin state	R	

24.13.3 Port L Inverting Register (PLIR)

PLIR is a 16-bit readable/writable register that enables or disables the inverting function for port L. Bits PL8IR to PL1IR correspond to pins PL8/TOE33 to PL1/TOE20/$\overline{\text{IRQ1}}$/$\overline{\text{POD}}$, respectively. The PLIR setting is valid regardless of the port L pin functions.

Setting the specific bits of PLIR to 1 inverts the output value on their corresponding pins.

PLIR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	PL8IR	PL7IR	PL6IR	PL5IR	PL4IR	PL3IR	PL2IR	PL1IR	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	Bit Name	Initial Value	R/W	Descriptions
15 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	PL8IR	0	R/W	These bits set whether or not the set value is inverted when it is output on a corresponding pin. 0: The set value is not inverted when output 1: The set value is inverted when output
7	PL7IR	0	R/W	
6	PL6IR	0	R/W	
5	PL5IR	0	R/W	
4	PL4IR	0	R/W	
3	PL3IR	0	R/W	
2	PL2IR	0	R/W	
1	PL1IR	0	R/W	
0	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

24.14 CK Control Register (CKCR)

CKCR is a 16-bit readable/writable register that enables or disables the CK output.

CKCR is initialized to H'0000 by a power-on reset, a transition to the hardware standby mode, or a power-on reset by the WDT. It is not initialized in sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	CKOE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Descriptions
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	CKOE	0	R/W	Enables or disables an output on the CK pin. 0: Enables output on the pin (CK output) 1: Disables output on the pin (Hi-Z)

24.15 Port Output Disable ($\overline{\text{POD}}$)

The output buffers of I/O port related pins (shown in table 24.13) can be controlled by the input level on the $\overline{\text{POD}}$ (port output disable) pin. This function is enabled regardless of the selected function when the corresponding pin is specified as an output. The output buffer control by the $\overline{\text{POD}}$ pin is performed asynchronously with the bus cycles.

Table 24.13 I/O Port Relating Pins

Relating Pins
PA11/A11/TOD13B/TIF2B to PA4/A4/TOD00B, PB11/$\overline{\text{CS1}}$/TOE21/TIF21/SSLB2, PB8/$\overline{\text{WAIT}}$/TOE20/TIF20/RxD_A, PC15/D15/TOD13A/SSLA3 to PC8/D8/TOD00A/TxD_A/CTx_A, PD13/TCLKB/TIJ1/TOE60 to PD0/TIOC00/TIOC31, PE13/TOE13 to PE6/TOE00/CTx_B to PE1/TIA01/TIOC42/TIOC40, PF15/TOD33B/CRx_B/RxD_A/FRRxD_B to PF0/TOD00B/TIF6, PG15/TOD33A/SSLA7/TIF13 to PG0/TOD00A/SSLA0, PL8/TOE33 to PL2/TOE21/$\overline{\text{IRQ2}}$

Table 24.14 $\overline{\text{POD}}$ Pin State

$\overline{\text{POD}}$	Descriptions
0	The pin output is disabled (high impedance)
1	The pin output is enabled (each setting function)

24.16 Usage Note

24.16.1 Note on State Immediately after Reset

Immediately after reset, input/output pins are in the input state. In this case, if the pins are at intermediate levels because, for example, they are opened, a passthrough current may flow. To reduce the passthrough current, set general input/output pins in the output state at low level, immediately after the LSI has been turned on. This countermeasure will set the input/output pins stable, and therefore the passthrough current will be reduced.

24.16.2 Note on Operation of Input/Output Pins on a Reset by an Internal Source

When input/output pins enter the reset state from the operating state due to a reset by an internal source, the states of the pins are undefined for up to one cycle of the peripheral clock (Pϕ). Over this period, the pins may be in the high-level output state, the low-level output state, or the high-impedance state.

Although the input/output pins are only temporarily in the undefined state described above, the microcontroller as a whole enters the specified reset state after that.

Resets by internal sources include internal resets by WDT overflow and H-UDI resets using emulators.

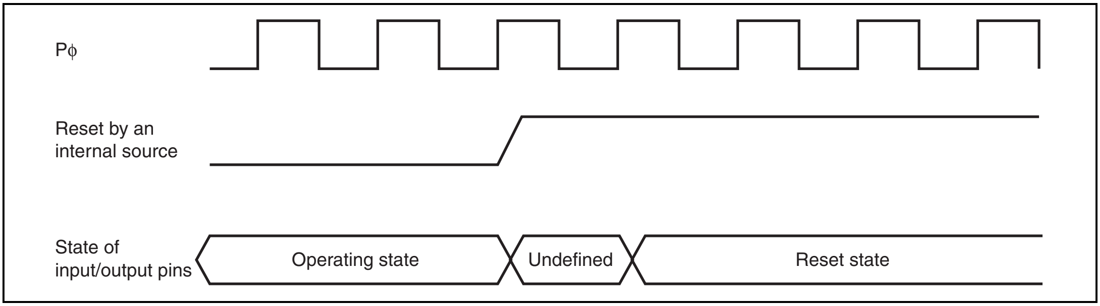


Figure 24.12 Operation of Input/Output Pins at Reset by Internal Source

Note: The phenomenon described above does not occur on any of dedicated input or output pins.

Resets by inputting a low level on the $\overline{\text{RES}}$ pin do not correspond the resets by internal sources.

Section 25 Multi-Input Signature Generator (MISG)

25.1 Overview

This LSI has a multi-input signature generator (MISG) for self test function.

The MISG monitors any write accesses by the CPU to the calculation data register (MISRCDR) and generates a 32-bit signature using the contents of MISRCDR. The signature is stored in the multi-input signature register (MISR). Each time MISRCDR is accessed, the MISG updates the contents of MISR with the two kinds of new signature which is generated by the data written to MISRCDR and the data stored in MISR (normally the data in MISR is the signature generated by the previous write access; in some cases, it is an initial value or a value that is directly written to MISR and MISR2).

Signatures are generated by the following polynomial.

MISR $G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x + 1$

MISR $G2(x) = x^{32} + x^{22} + x^{2} + x + 1$

The enable bit in the MISR control register (MISRCR) can be used to enable/disable the generation of signatures. In the initial status after a reset, signature generation is disabled so that writing to MISRCDR will not update the data stored in MISR and MISR2.

MISR and MISR2 are registers those are address-mapped to the internal I/O register space and can be directly read from or written to. A generated signature in MISR can be compared with an expected value or an arbitrary seed value (the value that is the source of a signature) can also be stored.

Figure 25.1 is a block diagram of the MISG.

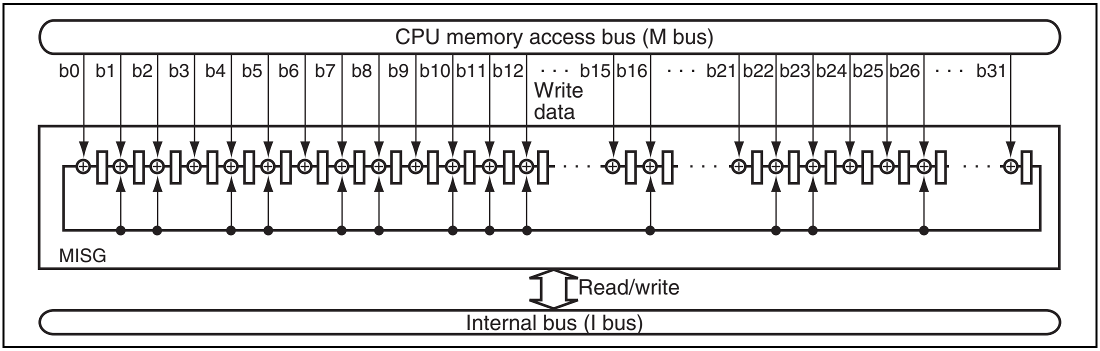

CPU memory access bus (M bus)
b0 b1 b2 b3 b4 b5 b6 b7 b8 b9 b10 b11 b12 · · · b15 b16 · · · b21 b22 b23 b24 b25 b26 · · · b31
Write data
MISG
Read/write
Internal bus (I bus)

Figure 25.1 Block Diagram of MISG (MISR)

Figure 25.2 shows the block diagram of MISG (MISR2).

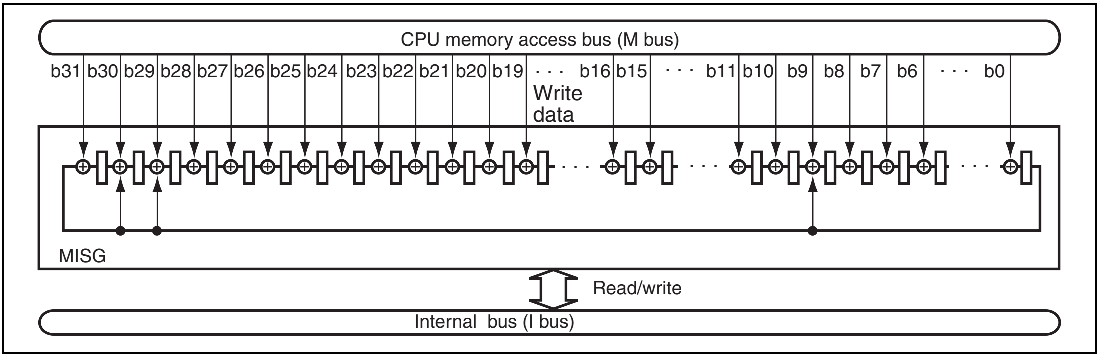

CPU memory access bus (M bus)
b31 b30 b29 b28 b27 b26 b25 b24 b23 b22 b21 b20 b19 · · · b16 b15 · · · b11 b10 b9 b8 b7 b6 · · · b0
Write data
MISG
Read/write
Internal bus (I bus)

Figure 25.2 Block Diagram of MISG (MISR2)

25.2 Register Descriptions

The MISG has the following registers.

Table 25.1 Register Configuration

Register Name	Symbol	R/W	Initial Value	Address	Access Size
Calculation data register	MISRCDR	W	—	H'FFF7FFFC	8, 16, 32
MISR control register	MISRCR	R/W	H'00	H'FFFC1C00	8
Multi-input signature register	MISR	R/W	H'00000000	H'FFFC1C04	32
Multi-input signature register2	MISR2	R/W	H'00000000	H'FFFC1C08	32

25.2.1 Calculation Data Register (MISRCDR)

MISRCDR is a 32-bit write-only register. MISRCDR can be written only by the CPU. MISRCDR can always be accessed in one write cycle (CPUCLK). The data written to MISRCDR is used as input data for the multi-input signature register (MISR).

Data can be written to MISRCDR in words, bytes, or longwords. However, the word or longword data is extended to 32-bit data by filling remaining bits with 0 so that 32-bit data is always input into MISR.

Note: Do not read from MISRCDR. Otherwise, correct operation cannot be guaranteed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MISRCD[31:16]															
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MISRCD[15:0]															
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	MISRCD [31:0]	—	W	Calculation Data Input data to MISR. Each time calculation data is written, a new signature is generated and it is stored in MISR.

25.2.2 Multi-Input Signature Register (MISR)

MISR is a 32-bit readable/writable register. Setting the MISR enable bit (MISREN) in the MISR control register (MISRCR) to 1 causes MISR to generate a new signature whenever data is written to MISRCDR, and the generated value is stored in MISR.

Signatures are generated by the following polynomial.

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x + 1$$

MISR is initialized to H'00000000 by a power-on reset or a transition to the hardware standby mode.

Note: When writing to MISRCDR after writing to this register, read this register before writing to MISRCDR. If MISR is not read, a correct signature may not be generated through a write operation to MISRCDR.

If the interval between writing to the MISRCDR register and reading from the MISR register is too short when four-times and two-times multiplication have been specified for the internal clock (ϕ) and peripheral clock (Pϕ), respectively, a signature generated by writing to MISRCDR may not be read correctly. If it is necessary to read from the MISR register immediately after writing to the MISRCDR register, execute three or more instructions before reading from the MISR register or read twice from the MISR register and use the second read data.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MISR[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MISR[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	MISR [31:0]	H'00000000	R/W	Signature Data These bits return the latest signature when read.

25.2.3 Multi-Input Signature Register 2(MISR2)

MISR2 is a 32-bit readable/writable register. Setting the MISR2 enable bit (MISR2EN) in the MISR control register (MISR2EN) to 1 causes MISR2 to generate a new signature whenever data is written to MISRCDR, and the generated value is stored in MISR2.

Signatures are generated by the following polynomial.

$$G(x) = x^{32} + x^{22} + x^{2} + x + 1$$

MISR2 is initialized to H'00000000 by a power-on reset or a transition to the hardware standby mode.

Note: When writing to MISRCDR after writing to this register, read this register before writing to MISRCDR. If MISR2 is not read, a correct signature may not be generated through a write operation to MISRCDR.

If the interval between writing to the MISRCDR register and reading from the MISR2 register is too short when four-times and two-times multiplication have been specified for the internal clock (ϕ) and peripheral clock (Pϕ), respectively, a signature generated by writing to MISRCDR may not be read correctly. If it is necessary to read from the MISR2 register immediately after writing to the MISRCDR register, execute three or more instructions before reading from the MISR2 register or read twice from the MISR2 register and use the second read data.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MISR2[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MISR2[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	MISR2 [31:0]	H'00000000	R/W	Signature Data These bits return the latest signature when read.

25.2.4 MISR Control Register (MISRCR)

MISRCR is an 8-bit readable/writable register that controls the generation of a signature by MISR. MISR and MISR2 generate a signature when the MISR enable bit (MISREN) is set to 1 and stores the resulting signature. If the MISREN bit is 0, the contents of MISR and MISR2 are not updated even when data is written to MISRCDR.

MISRCR is initialized to H'00 by a power-on reset or a transition to the hardware standby mode.

Note: When writing to MISRCDR after writing to this register, perform either of the following two operations before writing to MISRCDR. A correct signature may not be generated through a write operation to MISRCDR without performing either of them.

1 Read twice from the MISRCR register.

2 Execute three or more instructions after reading from the MISRCR register.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	MISR 2EN	MISREN
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	MISR2EN	0	R/W	MISR2 Enable 0: Disables generating a signature or updating the contents of MISR2 even when data is written to MISRCDR 1: Enables generating a signature or updating the contents of MISR2
0	MISREN	0	R/W	MISR Enable 0: Disables generating a signature or updating the contents of MISR even when data is written to MISRCDR 1: Enables generating a signature or updating the contents of MISR even when data is written to MISRCDR

Section 26 ROM

This LSI incorporates 4.0 Mbytes of flash memory (ROM) for the storage of instruction code. The flash memory has the following features.

26.1 Features

- Intelligent flash security

  The on-chip ROM has functionality for protection from programming and erasure (Intelligent Flash Security: IFS).

  For details, refer to section 27, Intelligent Flash Security (IFS).
- Two types of flash-memory MATs

  The ROM has two types of memory areas (hereafter referred to as memory MATs) in the same address space. These two MATs can be switched by the start-up mode or bank switching through the control register. For addresses H'00008000 to H'003FFFFF, undefined data is read and programming and erasing are ignored when the user boot MAT is selected.

  User MAT: 4.0 Mbytes

  User boot MAT: 32 Kbytes

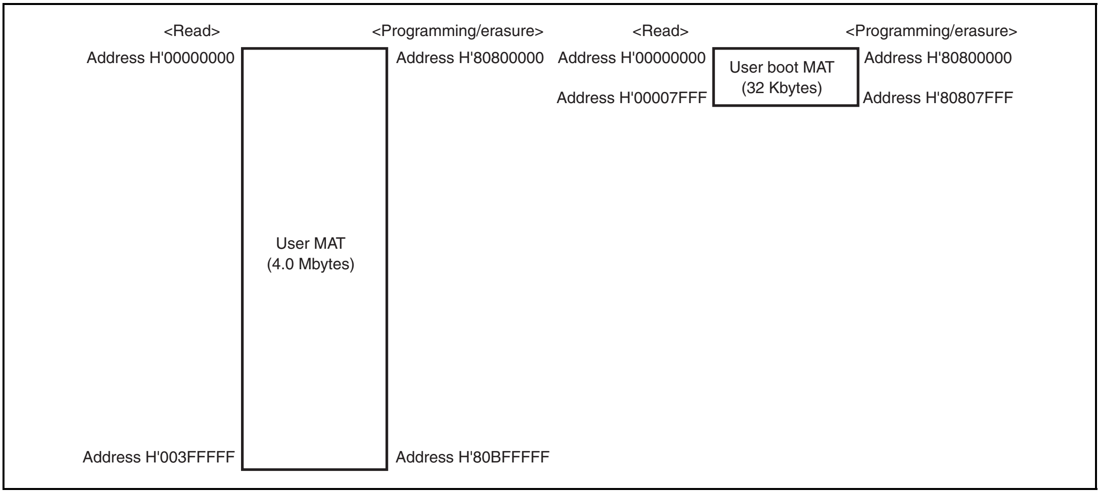


Figure 26.1 Memory MAT Configuration in ROM

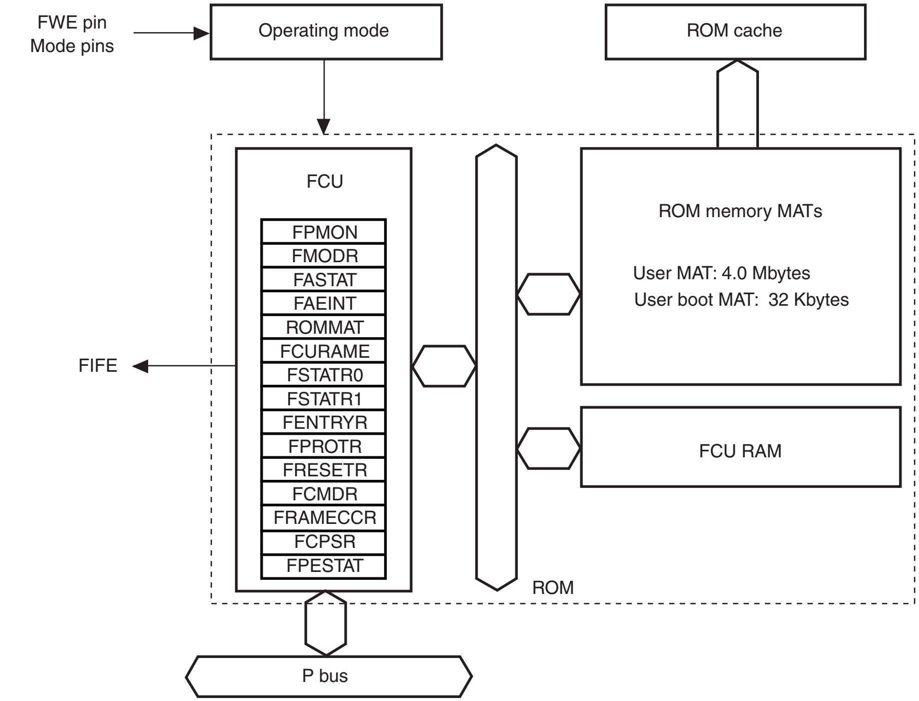
- High-speed reading through ROM cache

  Both the user MAT and user boot MAT can be read at high speed through the ROM cache. They can be read only in on-chip ROM enabled mode.

- Programming and erasing methods

  The ROM can be programmed and erased by commands issued through the peripheral bus (P bus) to the ROM/EEPROM-dedicated sequencer (FCU).

  While the flash control unit (FCU) is programming or erasing the ROM, the CPU can execute a program located outside the ROM. While the FCU is programming or erasing the EEPROM, the CPU can execute a program in the ROM. When the FCU suspends programming or erasure, the CPU can execute a program in the ROM, and then the FCU can resume programming or erasure. While the FCU suspends erasure, areas other than the erasure-suspended area can be programmed.



[Legend]

FPMON:	Flash pin monitor register
FMODR:	Flash mode register
FASTAT:	Flash access status register
FAEINT:	Flash access error interrupt enable register
ROMMAT:	ROM MAT select register
FCURAME:	FCU RAM enable register
FSTATR0, FSTATR1:	Flash status registers 0 and 1
FENTRYR:	Flash P/E mode entry register
FPROTR:	Flash protect register
FRESETR:	Flash reset register
FCMDR:	FCU command register
FCPSR:	FCU processing switch register
FRAMECCR:	FCU RAM ECC error control register
FIFE:	Flash interface error interrupt
FPESTAT:	Falsh P/E status register

Figure 26.2 Block Diagram of ROM

- Programming/erasing unit

  The user MAT and user boot MAT are programmed in 256-byte units. The entire area of the user boot MAT is always erased at one time. The user MAT can be erased in block units.

  Figure 26.3 shows the block configuration of the user MAT. The user MAT is divided into eight 8-Kbyte blocks, nine 64-Kbyte blocks, and twenty-seven 128-Kbyte blocks.

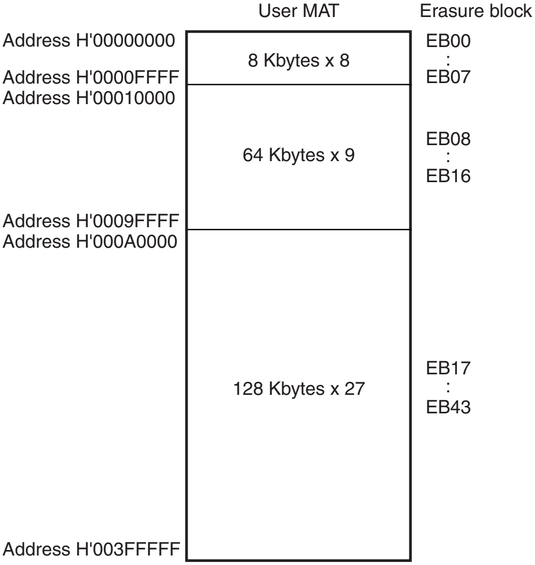


Figure 26.3 Block Configuration of User MAT

- Three types of on-board programming modes
  - Boot mode

    The user MAT and user boot MAT can be programmed using the SCI. The bit rate for SCI communications between the host and this LSI can be automatically adjusted.
  - User program mode

    The user MAT can be programmed with a desired interface. A transition from MCU mode 2 (MCU extended mode) or mode 3 (MCU single-chip mode) to this mode is enabled simply by changing the level on the FWE pin.
  - User boot mode

    The user MAT can be programmed with a desired interface. To make a transition to this mode, a reset is needed.
- Protection modes

  This LSI supports two modes to protect memory against programming or erasure: hardware protection by the levels on the FWE and mode pins and software protection by the FENTRY6 to FENTRY0 bits or lock bit settings. The FENTRY6 to FENTRY0 bits enable or disable ROM programming or erasure by the FCU. A lock bit is included in each erasure block of the user MAT to protect memory against programming or erasure.

  The LSI also provides a function to suspend programming or erasure when abnormal operation is detected during programming or erasure.
- Programming and erasing time and count

  Refer to section 34, Electrical Characteristics.

26.2 Input/Output Pins

Table 26.1 shows the input/output pins used for the ROM. The combination of MD4 to MD0 pin levels and the FWE pin level determines the ROM programming mode (see section 26.4, Overview of ROM-Related Modes). In boot mode, the ROM can be programmed or erased by the host connected via the PJ6/RxD_A and PJ5/TxD_A pins (see section 26.5, Boot Mode).

Table 26.1 Pin Configuration

Pin Name	Symbol	I/O	Function
Power-on reset	$\overline{\text{RES}}$	Input	This LSI enters the power-on reset state when this signal goes low.
Mode	MD4 to MD0	Input	These pins specify the operating mode.
Flash programming enable	FWE	Input	This pin enables or disables ROM programming.
Receive data in SCI channel A	PJ6/RxD_A	Input	Receives data through SCI channel A (communications with host)
Transmit data in SCI channel A	PJ5/TxD_A	Output	Transmits data through SCI channel A (communications with host)

26.3 Register Descriptions

Table 26.2 shows the ROM-related registers. Some of these registers have EEPROM-related bits, but this section only describes the ROM-related bits. For the EEPROM-related bits, refer to section 28.3, Register Descriptions. The ROM-related registers are initialized by a power-on reset or a transition to the hardware standby mode.

Table 26.2 Register Configuration

Register Name	Symbol	R/W*[1]	Initial Value	Address	Access Size
Flash pin monitor register	FPMON	R	H'00 H'80	H'FFFFA800	8
Flash mode register	FMODR	R/W	H'00	H'FFFFA802	8
Flash access status register	FASTAT	R/(W)*[2]	H'00	H'FFFFA810	8
Flash access error interrupt enable register	FAEINT	R/W	H'9F	H'FFFFA811	8
ROM MAT select register	ROMMAT	R/(W)*[3]	H'0000 H'0001	H'FFFFA820	8, 16
FCU RAM enable register	FCURAME	R/(W)*[3]	H'0000	H'FFFFA854	8, 16
Flash status register 0	FSTATR0	R	H'80*[5]	H'FFFFA900	8, 16
Flash status register 1	FSTATR1	R	H'00*[5]	H'FFFFA901	8, 16
Flash P/E mode entry register	FENTRYR	R/(W)*[4]	H'0000*[5]	H'FFFFA902	8, 16
Flash protect register	FPROTR	R/(W)*[4]	H'0000*[5]	H'FFFFA904	8, 16
Flash reset register	FRESETR	R/(W)*[3]	H'0000	H'FFFFA906	8, 16
FCU command register	FCMDR	R	H'FFFF*[5]	H'FFFFA90A	8, 16
FCU RAM ECC error control register	FRAMECCR	R/W	H'02*[5]	H'FFFFA90C	8
FCU processing switch register	FCPSR	R/W	H'0000*[5]	H'FFFFA918	8, 16
Flash P/E status register	FPESTAT	R	H'0000*[5]	H'FFFFA91C	8, 16
IFS control mode register	IFSCMR	R/(W)*[3]	H'0000	H'FFFFA890	8, 16
IFS status register	IFSSR	R	H'00	H'FFFFA912	8, 16

Notes: 1. In on-chip ROM disabled mode, the ROM-related registers are always read as 0 and writing to them is ignored.

2. This register consists of the bits where only 0 can be written to clear the flags and the read-only bits.

3. This register can be written to only when a specified value is written to the upper byte in word access. The data written to the upper byte is not stored in the register.
4. This register can be written to only when a specified value is written to the upper byte in word access; the register is initialized when a value not allowed for the register is written to the upper byte. The data written to the upper byte is not stored in the register.
5. These registers can be initialized by a power-on reset, a transition to the hardware standby mode, or setting the FRESET bit of FRESETR to 1.

26.3.1 Flash Pin Monitor Register (FPMON)

FPMON monitors the FWE pin state. FPMON is read as H'00 in on-chip ROM disabled mode. FPMON is initialized by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	FWE	—	—	—	—	—	—	—
Initial value:	1/0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	FWE	1/0	R	Flash Write Enable Monitors the FWE pin level. The initial value depends on the FWE pin level when the LSI is started. 0: Disables ROM programming and erasure 1: Enables ROM programming and erasure
6 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

26.3.2 Flash Mode Register (FMODR)

FMODR specifies the FCU operation mode. In on-chip ROM disabled mode, FMODR is read as H'00 and writing to it is ignored. FMODR is initialized by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	FRDMD	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	FRDMD	0	R/W	FCU Read Mode Select Selects the read mode to read the ROM or EEPROM using FCU. This bit specifies the check method for the lock bits in the ROM (see section 26.6.1, FCU Command List, and section 26.6.3 (11), Reading Lock Bit), whereas this bit must be set to make the blank check command available for use in the EEPROM (see section 28, EEPROM). 0: Selects the memory area read mode. The mode to read the lock bits in the ROM in ROM lock bit read mode. 1: Selects the register read mode. The mode to read the lock bits in the ROM using the lock bit read 2 command.
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

26.3.3 Flash Access Status Register (FASTAT)

FASTAT indicates the access error status for the ROM and EEPROM. In on-chip ROM disabled mode, FASTAT is read as H'00 and writing to it is ignored. If any bit in FASTAT is set to 1, the FCU enters command-locked state (see section 26.8.3, Error Protection). To cancel a command-locked state, set FASTAT to H'10, and then issue a status-clear command to the FCU. FASTAT is initialized by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	ROMAE	—	—	CMDLK	EEPAE	EEPIFE	EEPRPE	EEPWPE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only 0 can be written to clear the flag after 1 is read.

Bit	Bit Name	Initial Value	R/W	Description
7	ROMAE	0	R/(W)*	Access Error Indicates whether or not a ROM access error has been generated. If this bit becomes 1, the ILGLERR bit in FSTATR0 is set to 1 and the FCU enters a command-locked state. 0: No ROM access error has occurred. 1: A ROM access error has occurred. [Setting conditions] • A read access is issued to ROM program/erase addresses H'80BC0000 to H'80BFFFFF while the FENTRY6 bit in FENTRYR is 1 in ROM P/E normal mode. • A read access is issued to ROM program/erase addresses H'80B80000 to H'80BBFFFF while the FENTRY5 bit in FENTRYR is 1 in ROM P/E normal mode. • A read access command is issued to ROM program/erase addresses H'80B40000 to H'80B7FFFF while the FENTRY4 bit in FENTRYR is 1 in ROM P/E normal mode. • A read access command is issued to ROM program/erase addresses H'80B00000 to H'80B3FFFF while the FENTRY3 bit in FENTRYR is in ROM P/E normal mode.

Bit	Bit Name	Initial Value	R/W	Description
7	ROMAE	0	R/(W)*	• A read access is performed to ROM program/erase addresses H'80A00000 to H'80AFFFFF while the FENTRY2 bit in FENTRYR is 1 in ROM P/E normal mode. • A read access command is issued to ROM program/erase addresses H'80900000 to H'809FFFFF while the FENTRY1 bit in FENTRYR is 1 in ROM P/E normal mode. • A read access command is issued to ROM program/erase addresses H'80800000 to H'808FFFFF while the FENTRY0 bit in FENTRYR is 1 in ROM P/E normal mode. • An access is issued to ROM program/erase addresses H'80BC0000 to H'80BFFFFF while the FENTRY6 bit in FENTRYR is 0. • An access is issued to ROM program/erase addresses H'80B80000 to H'80BBFFFF while the FENTRY5 bit in FENTRYR is 0. • An access command is issued to ROM program/erase addresses H'80B40000 to H'80B7FFFF while the FENTRY4 bit in FENTRYR is 0. • An access command is issued to ROM program/erase addresses H'80B00000 to H'80B3FFFF while the FENTRY3 bit in FENTRYR is 0. • An access is issued to ROM program/erase addresses H'80A00000 to H'80AFFFFF while the FENTRY2 bit in FENTRYR is 0. • An access command is issued to ROM program/erase addresses H'80900000 to H'809FFFFF while the FENTRY1 bit in FENTRYR is 0. • An access command is issued to ROM program/erase addresses H'80800000 to H'808FFFFF while the FENTRY0 bit in FENTRYR is 0.

Bit	Bit Name	Initial Value	R/W	Description
7	ROMAE	0	R/(W)*	• A read access command is issued to ROM read addresses H'00000000 to H'003BFFFF while the FENTRYR register value is not H'0000. • A block erase, program, or lock bit program command is issued while the user boot MAT is selected. • An access command is issued to an address other than ROM program/erase addresses H'80800000 to H'80807FFF while the user boot MAT is selected. [Clearing condition] A 0 is written to this bit after reading a 1 from the ROMAE bit.
6, 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	CMDLK	0	R	FCU Command Lock Indicates whether the FCU is in command-locked state (see section 26.8.3, Error Protection). 0: The FCU is not in a command-locked state 1: The FCU is in a command-locked state [Setting condition] • The FCU detects an error and enters command-locked state. [Clearing condition] • The FCU completes the status-clear command processing while FASTAT is H'10.
3	EEPAE	0	R/(W)*	EEPROM Access Error Refer to section 28, EEPROM.
2	EEPIFE	0	R/(W)*	EEPROM Instruction Fetch Error Refer to section 28, EEPROM.
1	EEPRPE	0	R/(W)*	EEPROM Read Protect Error Refer to section 28, EEPROM.
0	EEPWPE	0	R/(W)*	EEPROM Program/Erase Protect Error Refer to section 28, EEPROM.

Note: * Writing a 0 after reading a 1 is only allowed in order to clear the flag.

26.3.4 Flash Access Error Interrupt Enable Register (FAEINT)

FAEINT enables or disables output of flash interface error (FIFE) interrupts. In on-chip ROM disabled mode, FAEINT is read as H'00 and writing to it is ignored. FAEINT is initialized by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	ROM AEIE	—	—	CMD LKIE	EEP AEIE	EEPI FEIE	EEPR PEIE	EEPW PEIE
Initial value:	1	0	0	1	1	1	1	1
R/W:	R/W	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	ROMAEIE	1	R/W	ROM Access Error Interrupt Enable Enables or disables an FIFE interrupt request when a ROM access error occurs and the ROMAE bit in FASTAT becomes 1. 0: Does not generate an FIFE interrupt request when ROMAE = 1. 1: Generates an FIFE interrupt request when ROMAE = 1.
6, 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
4	CMDLKIE	1	R/W	FCU Command Lock Interrupt Enable Enables or disables an FIFE interrupt request when FCU command-locked state is entered and the CMDLK bit in FASTAT becomes 1. 0: Does not generate an FIFE interrupt request when CMDLK = 1 1: Generates an FIFE interrupt request when CMDLK = 1
3	EEPAEIE	1	R/W	EEPROM Access Error Interrupt Enable Refer to section 28, EEPROM.
2	EEPIFEIE	1	R/W	EEPROM Instruction Fetch Error Interrupt Enable Refer to section 28, EEPROM.
1	EEPRPEIE	1	R/W	EEPROM Read Protect Error Interrupt Enable Refer to section 28, EEPROM.
0	EEPWPEIE	1	R/W	EEPROM Program/Erase Protect Error Interrupt Enable Refer to section 28, EEPROM.

26.3.5 ROM MAT Select Register (ROMMAT)

ROMMAT switches memory MATs in the ROM. In on-chip ROM disabled mode, ROMMAT is read as H'0000 and writing to it is ignored. ROMMAT is initialized by a power-on reset or a transition to the hardware standby mode.

When switching MAT by changing ROMMAT register setting, write to ROMMAT register and execute a minimum of 5 NOP instructions after performing a dummy read of ROMMAT register.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KEY								—	—	—	—	—	—	—	ROM SEL
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R	R	R	R	R	R/W

Note: * Written data is not stored in these bits.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	KEY	H'00	R/(W)*	Key Code These bits enable or disable ROMSEL bit modification. The data written to these bits are not stored.
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	ROMSEL	0/1	R/W	ROM MAT Select Selects a memory MAT in the ROM. The initial value is 1 when the LSI is started in user boot mode; otherwise, the initial value is 0. Writing to this bit is enabled only when this register is accessed in word size and H'3B is written to the KEY bits. 0: Selects the user MAT 1: Selects the user boot MAT

Note: Write data is not retained.

26.3.6 FCU RAM Enable Register (FCURAME)

FCURAME enables or disables access to the FCU RAM area. In on-chip ROM disabled mode, FCURAME is read as H'00 and writing to it is ignored. FCURAME is initialized by a power-on reset or a transition to the hardware standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KEY								—	—	—	—	—	—	—	FCRME
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R	R	R	R	R	R/W

Note: * Written data is not stored in these bits.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	KEY	H'00	R/(W)*	Key Code These bits enable or disable FCRME bit modification. The data written to these bits are not stored.
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	FCRME	0	R/W	FCU RAM Enable Enables or disables access to the FCU RAM. Writing to this bit is enabled only when this register is accessed in word size and H'C4 is written to the KEY bits. Before writing to the FCU RAM, clear FENTRYR to H'0000 to stop the FCU. 0: Disables access to FCU RAM 1: Enables access to FCU RAM

Note: Write data is not retained.

26.3.7 Flash Status Register 0 (FSTATR0)

FSTATR0 indicates the FCU status. In on-chip ROM disabled mode, FSTATR0 is read as H'00. FRTATR0 is initialized by a power-on reset, a transition to the hardware standby mode, or setting the FRESET bit of the FRESETR register is set to 1.

Bit:	7	6	5	4	3	2	1	0
	FRDY	ILG LERR	ERS ERR	PRG ERR	SUS RDY	—	ERS SPD	PRG SPD
Initial value:	1	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	FRDY	1	R	Flash Ready Indicates the processing state in the FCU. 0: Programming or erasure processing, programming or erasure suspension processing, lock bit read 2 command processing, or EEPROM blank check is in progress (see section 28, EEPROM). 1: None of the above is in progress.
6	ILGLERR	0	R	Illegal Command Error Indicates that the FCU has detected an illegal command or illegal ROM or EEPROM access. When this bit is 1, the FCU is in command-locked state (see section 26.8.3, Error Protection). 0: The FCU has not detected any illegal command or illegal ROM/EEPROM access 1: The FCU has detected an illegal command or illegal ROM/EEPROM access [Setting conditions] • The FCU has detected an illegal command. • The FCU has detected an illegal ROM/EEPROM access (the ROMAE, EEPAE, EEPIFE, EEPRPE, or EEPWPE bit in FASTAT is 1). • The FENTRYR setting is illegal. [Clearing condition] • The FCU completes the status-clear command processing while FASTAT is H'10.

Bit	Bit Name	Initial Value	R/W	Description
5	ERSERR	0	R	Erasure Error Indicates the result of ROM or EEPROM erasure by the FCU. When this bit is 1, the FCU is in command-locked state (see section 26.8.3, Error Protection). 0: Erasure processing has been completed successfully 1: An error has occurred during erasure [Setting conditions] • An error has occurred during erasure. • A block erase command has been issued for the area protected by a lock bit. [Clearing condition] • The FCU completes the status-clear command processing.
4	PRGERR	0	R	Programming Error Indicates the result of ROM or EEPROM programming by the FCU. When this bit is 1, the FCU is in command-locked state (see section 26.8.3, Error Protection). 0: Programming has been completed successfully 1: An error has occurred during programming [Setting conditions] • An error has occurred during programming. • A programming command has been issued for the area protected by a lock bit. [Clearing condition] • The FCU completes the status-clear command processing.

Bit	Bit Name	Initial Value	R/W	Description
3	SUSRDY	0	R	Suspend Ready Indicates whether the FCU is ready to accept a P/E suspend command. 0: The FCU cannot accept a P/E suspend command 1: The FCU can accept a P/E suspend command [Setting condition] • After initiating programming/erasure, the FCU has entered a state where it is ready to accept a P/E suspend command. [Clearing conditions] • The FCU has accepted a P/E suspend command. • The FCU has entered a command-locked state during programming or erasure.
2	—	0	R	Reserved This bit is always read as 0. Correct operation is not guaranteed if 1 is written to this bit.
1	ERSSPD	0	R	Erasure-Suspended Status Indicates that the FCU has entered an erasure suspension process or an erasure-suspended status (see section 26.6.4, Suspending Operation). 0: The FCU is in a status other than the below-mentioned. 1: The FCU is in an erasure suspension process or an erasure-suspended status. [Setting condition] • The FCU has initiated an erasure suspend command. [Clearing condition] • The FCU has accepted a resume command.

Bit	Bit Name	Initial Value	R/W	Description
0	PRGSPD	0	R	Programming-Suspended Status Indicates that the FCU has entered a write suspension process or a write suspend status (see section 26.6.4, Suspending Operation). 0: The FCU is in a status other than the below-mentioned. 1: The FCU is in a write suspension process or a write-suspended status. [Setting condition] • The FCU has initiated a write suspend command. [Clearing condition] • The FCU has accepted a resume command.

26.3.8 Flash Status Register 1 (FSTATR1)

FSTATR1 indicates the FCU status. In on-chip ROM disabled mode, FSTATR1 is read as H'00. FSTATR1 is initialized by a power-on reset, a transition to the hardware standby mode, or setting the FRESET bit of the FRESETR register is set to 1.

Bit:	7	6	5	4	3	2	1	0
	FCU ERR	—	—	FLO CKST	—	—	FRD TCT	FRC RCT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	FCUERR	0	R	FCU Error Indicates an error has occurred during the CPU processing in the FCU. 0: No error has occurred during the CPU processing in the FCU 1: An error has occurred during the CPU processing in the FCU [Clearing condition] • The FRESET bit in FRESETR is set to 1. When FCUERR is 1, set the FRESET bit to 1 to initialize the FCU, and then copy the FCU firmware again from the FCU firmware area to the FCU RAM area.
6, 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	FLOCKST	0	R	Lock Bit Status Reflects the lock bit data read through lock bit read 2 command execution. When the FRDY bit becomes 1 after the lock bit read 2 command is issued, valid data is stored in this bit. This bit value is retained until the next lock bit read 2 command is completed. 0: Protected state 1: Non-protected state

Bit	Bit Name	Initial Value	R/W	Description
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	FRDTCT	0	R	FCU RAM ECC 2-Bit Error Detection Monitoring Bit Indicates that a 2-bit error has been detected when the FCU is reading RAM. 0: No 2-bit error has been detected. 1: A 2-bit error has been detected. When FRDTCT is 1, set the FRESET bit to 1 to initialize the FCU, and then copy the FCU firmware again from the FCU firmware area to the FCU RAM area.
0	FRCRCT	0	R	FCU RAM ECC 1-Bit Error Correction Monitoring Bit Indicates that a 1-bit error has been corrected when the FCU is reading RAM. 0: No 1-bit error has been corrected. 1: A 1-bit error has been corrected. When FRCRCT is 1, set the FRESET bit to 1 to initialize the FCU, and then copy the FCU firmware again from the FCU firmware area to the FCU RAM area.

26.3.9 FCU RAM ECC Error Control Register (FRAMECCR)

FRAMECCR enables or disables an FCU command lock request upon correction of an ECC 1-bit error or detection of an ECC 2-bit error when the FCU is reading the RAM. FRAMECCR enables or disables an FCU command lock request, but it does not control the FRDTCT and FRCRCT bits of the flash status register 1 (FSTATR1). In on-chip ROM disabled mode, FRAMECCR is read as H'00 and writing to it is ignored. FRAMECCR is initialized by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	FRD CLE	FRC CLE
Initial value:	0	0	0	0	0	0	1	0
R/W:	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	FRDCLE	1	R/W	FCU Command Lock Enabling Bit upon FCU RAM 2-Bit Error Detection Enables or disables an FCU command-lock request upon detection of an ECC 2-bit error when the FCU is reading RAM. If an ECC 2-bit error is detected while this bit is set to 1, the CMDLK bit of FASTAT is set to 1. 0: Issues no FCU command-lock request upon detection of a 2-bit error (setting prohibited). 1: Issues an FCU command-lock request upon detection of a 2-bit error.
0	FRCCLE	0	R/W	FCU Command Lock Enabling Bit upon FCU RAM 1-Bit Error Correction Enables or disables an FCU command-lock request upon correction of an ECC 1-bit error when the FCU is reading RAM. If an ECC 1-bit error is detected while this bit is set to 1, the CMDLK bit of FASTAT is set to 1. 0: Issues no FCU command-lock request upon detection of a 1-bit error. 1: Issues an FCU command-lock request upon detection of a 1-bit error.

26.3.10 Flash P/E Mode Entry Register (FENTRYR)

FENTRYR specifies the P/E mode for the ROM or EEPROM. To specify the P/E mode for the ROM or EEPROM so that the FCU can accept commands, set any one of the FENTRYD, FENTRY6, FENTRY5, FENTRY4, FENTRY3, FENTRY2, FENTRY1, or FENTRY0 bit to 1. Note that if this register is set to other than H'0001, H'0002, H'0004, H'0008, H'0010, H'0020, H'0040, and H'0080, the ILGLERR bit in the FSTATR0 register will be set and the FCU will enter command-locked state. In on-chip ROM disabled mode, FENTRYR is read as H'0000 and writing to it is ignored. FENTRYR can be initialized by a power-on reset, a transition to the hardware standby mode, or setting the FRESET bit of FRESETR to 1.

When transiting to ROM read mode by changing FENTRY6 to FENTRY0 bit in FENTRYR register from 1 to 0, write 0 to FENTRY6 to FENTRY0 bit and execute a minimum of 5 NOP instructions after performing a dummy read of FENTRYR register.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FEKEY								FEN TRYD	FEN TRY6	FEN TRY5	FEN TRY4	FEN TRY3	FEN TRY2	FEN TRY1	FEN TRY0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Notes: * Written data is not stored in these bits.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	FEKEY	All 0	R/(W)*	Key Code These bits enable or disable rewriting of the FENTRYD and FENTRY6 to FENTRY0 bits. Write data to these bits are not retained.
7	FENTRYD	0	R/W	EEPROM P/E Mode Entry Bit Refer to section 28, EEPROM.

Bit	Bit Name	Initial Value	R/W	Description
6	FENTRY6	0	R/W	ROM P/E Mode Entry Bit 6 Specifies the P/E mode for the 0.25-Mbyte ROM (read addresses: H'003C0000 to H'003FFFFF; program/erase addresses: H'80BC0000 to H'80BFFFFF). 0: The 0.25-Mbyte ROM is in read mode. 1: The 0.25-Mbyte ROM is in P/E mode. Programming is enabled when the following conditions are all satisfied: • The LSI is in on-chip ROM enabled mode. • The FWE bit in FPMON is 1. • The FRDY bit in FSTATR0 is 1. • H'AA is written to FEKEY in word access. [Setting condition] • 1 is written to FENTRY6 while the write enabling conditions are satisfied and FENTRYR is H'0000. [Clearing conditions] • The FRDY bit in FSTATR0 becomes 1 and the FWE bit in FPMON becomes 0. • This register is written to in byte access. • A value other than H'AA is written to FEKEY in word access. • 0 is written to FENTRY6 while the write enabling conditions are satisfied. • FENTRYR is written to while FENTRYR is not H'0000 and the write enabling conditions are satisfied.

Bit	Bit Name	Initial Value	R/W	Description
5	FENTRY5	0	R/W	ROM P/E Mode Entry Bit 5 Specifies the P/E mode for the 0.25-Mbyte ROM (read addresses: H'00380000 to H'003BFFFF; program/erase addresses: H'80B80000 to H'80BBFFFF). 0: The 0.25-Mbyte ROM is in read mode. 1: The 0.25-Mbyte ROM is in P/E mode. Programming is enabled when the following conditions are all satisfied: • The LSI is in on-chip ROM enabled mode. • The FWE bit in FPMON is 1. • The FRDY bit in FSTATR0 is 1. • H'AA is written to FEKEY in word access. [Setting condition] • 1 is written to FENTRY5 while the write enabling conditions are satisfied and FENTRYR is H'0000. [Clearing conditions] • The FRDY bit in FSTATR0 becomes 1 and the FWE bit in FPMON becomes 0. • This register is written to in byte access. • A value other than H'AA is written to FEKEY in word access. • 0 is written to FENTRY5 while the write enabling conditions are satisfied. • FENTRYR is written to while FENTRYR is not H'0000 and the write enabling conditions are satisfied.

Bit	Bit Name	Initial Value	R/W	Description
4	FENTRY4	0	R/W	ROM P/E Mode Entry Bit 4 Specifies the P/E mode for the 0.25-Mbyte ROM (read addresses: H'00340000 to H'0037FFFF; program/erase addresses: H'80B40000 to H'80B7FFFF). 0: The 0.25-Mbyte ROM is in read mode. 1: The 0.25-Mbyte ROM is in P/E mode. Programming is enabled when the following conditions are all satisfied: • The LSI is in on-chip ROM enabled mode. • The FWE bit in FPMON is 1. • The FRDY bit in FSTATR0 is 1. • H'AA is written to FEKEY in word access. [Setting condition] • 1 is written to FENTRY4 while the write enabling conditions are satisfied and FENTRYR is H'0000. [Clearing conditions] • The FRDY bit in FSTATR0 becomes 1 and the FWE bit in FPMON becomes 0. • This register is written to in byte access. • A value other than H'AA is written to FEKEY in word access. • 0 is written to FENTRY4 while the write enabling conditions are satisfied. • FENTRYR is written to while FENTRYR is not H'0000 and the write enabling conditions are satisfied.

Bit	Bit Name	Initial Value	R/W	Description
3	FENTRY3	0	R/W	ROM P/E Mode Entry Bit 3 Specifies the P/E mode for the 0.25-Mbyte ROM (read addresses: H'00300000 to H'0033FFFF; program/erase addresses: H'80B00000 to H'80B3FFFF). 0: The 0.25-Mbyte ROM is in read mode. 1: The 0.25-Mbyte ROM is in P/E mode. Programming is enabled when the following conditions are all satisfied: • The LSI is in on-chip ROM enabled mode. • The FWE bit in FPMON is 1. • The FRDY bit in FSTATR0 is 1. • H'AA is written to FEKEY in word access. [Setting condition] • 1 is written to FENTRY3 while the write enabling conditions are satisfied and FENTRYR is H'0000. [Clearing conditions] • The FRDY bit in FSTATR0 becomes 1 and the FWE bit in FPMON becomes 0. • This register is written to in byte access. • A value other than H'AA is written to FEKEY in word access. • 0 is written to FENTRY3 while the write enabling conditions are satisfied. • FENTRYR is written to while FENTRYR is not H'0000 and the write enabling conditions are satisfied.

Bit	Bit Name	Initial Value	R/W	Description
2	FENTRY2	0	R/W	ROM P/E Mode Entry Bit 2 These bits specify the P/E mode for the 1-Mbyte ROM (read addresses: H'00200000 to H'002FFFFF; program/erase addresses: H'80A00000 to H'80AFFFFF). 0: the 1-Mbyte ROM is in read mode 1: the 1-Mbyte ROM is in P/E mode Programming is enabled when the following conditions are all satisfied: • The LSI is in on-chip ROM enabled mode. • The FWE bit in FPMON is 1. • The FRDY bit in FSTATR0 is 1. • H'AA is written to FEKEY in word access. [Setting condition] • 1 is written to FENTRY2 while the write enabling conditions are satisfied and FENTRYR is H'0000. [Clearing conditions] • The FRDY bit in FSTATR0 becomes 1 and the FWE bit in FPMON becomes 0. • This register is written to in byte access. • A value other than H'AA is written to FEKEY in word access. • 0 is written to FENTRY2 while the write enabling conditions are satisfied. • FENTRYR is written to while FENTRYR is not H'0000 and the write enabling conditions are satisfied.

Bit	Bit Name	Initial Value	R/W	Description
1	FENTRY1	0	R/W	ROM P/E Mode Entry Bit 1 These bits specify the P/E mode for the 1-Mbyte ROM (read addresses: H'00100000 to H'001FFFFF; program/erase addresses: H'80900000 to H'809FFFFF). 0: the 1-Mbyte ROM is in read mode 1: the 1-Mbyte ROM is in P/E mode Programming is enabled when the following conditions are all satisfied: • The LSI is in on-chip ROM enabled mode. • The FWE bit in FPMON is 1. • The FRDY bit in FSTATR0 is 1. • H'AA is written to FEKEY in word access. [Setting condition] • 1 is written to FENTRY while the write enabling conditions are satisfied and FENTRYR is H'0000. [Clearing conditions] • The FRDY bit in FSTATR0 becomes 1 and the FWE bit in FPMON becomes 0. • This register is written to in byte access. • A value other than H'AA is written to FEKEY in word access. • 0 is written to FENTRY while the write enabling conditions are satisfied. • FENTRYR is written to while FENTRYR is not H'0000 and the write enabling conditions are satisfied.

Bit	Bit Name	Initial Value	R/W	Description
0	FENTRY0	0	R/W	ROM P/E Mode Entry Bit 0 These bits specify the P/E mode for the 1-Mbyte ROM (read addresses: H'00000000 to H'000FFFFF; program/erase addresses: H'80800000 to H'808FFFFF). 0: the 1-Mbyte ROM is in read mode 1: the 1-Mbyte ROM is in P/E mode Programming is enabled when the following conditions are all satisfied: • The LSI is in on-chip ROM enabled mode. • The FWE bit in FPMON is 1. • The FRDY bit in FSTATR0 is 1. • H'AA is written to FEKEY in word access. [Setting condition] • 1 is written to FENTRY while the write enabling conditions are satisfied and FENTRYR is H'0000. [Clearing conditions] • The FRDY bit in FSTATR0 becomes 1 and the FWE bit in FPMON becomes 0. • This register is written to in byte access. • A value other than H'AA is written to FEKEY in word access. • 0 is written to FENTRY while the write enabling conditions are satisfied. • FENTRYR is written to while FENTRYR is not H'0000 and the write enabling conditions are satisfied.

Note: * Write data is not retained.

26.3.11 Flash Protect Register (FPROTR)

FPROTR enables or disables the protection function through the lock bits against programming and erasure. In on-chip ROM disabled mode, FPROTR is read as H'0000 and writing to it is ignored. FPROTR is initialized by a power-on reset, a transition to the hardware standby mode, or setting the FRESET bit of FRESETR to 1.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FPKEY								—	—	—	—	—	—	—	FPROTCN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R	R	R	R	R	R/W

Note: * Written data is not stored in these bits.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	FPKEY	H'00	R/(W)*	Key Code These bits enable or disable FPROTCN bit modification. The data written to these bits are not stored.
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	FPROTCN	0	R/W	Lock Bit Protect Cancel Enables or disables protection through the lock bits against programming and erasure. 0: Enables protection through the lock bits 1: Disables protection through the lock bits [Setting condition] • H'55 is written to FPKEY and 1 is written to FPROTCN in word access while the FENTRYR register value is not H'0000. [Clearing conditions] • This register is written to in byte access. • A value other than H'55 is written to FPKEY in word access. • H'55 is written to FPKEY and 0 is written to FPROTCN in word access. • The FENTRYR register value is H'0000.

Note: * Write data is not retained.

26.3.12 Flash Reset Register (FRESETR)

FRESETR is used for the initialization of FCU. In on-chip ROM disabled mode, FRESETR is read as H'0000 and writing to it is ignored. FRESETR is initialized by a power-on reset or a transition to the hardware standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FRKEY								—	—	—	—	—	—	—	FRESET
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R	R	R	R	R	R/W

Note: * Written data is not stored in these bits.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	FRKEY	H'00	R/(W)*	Key Code These bits enable or disable FRESET bit modification. The data written to these bits are not stored.
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	FRESET	0	R/W	Flash Reset Setting this bit to 1 forcibly terminates programming/erasure of ROM or EEPROM and initializes the FCU. A high voltage is applied to the ROM/EEPROM memory units during programming and erasure. To ensure sufficient time for the voltage applied to the memory unit to drop, keep the value of the FRESET bit at 1 for a period of t_{RESW2} (see section 34, Electrical Characteristics) when the FCU is initialized. Do not read from the ROM/EEPROM units while the value of the FRESET bit is kept at 1. The FCU commands are unavailable for use while the FRESET bit is set to 1, since this initializes the FENTRYR register. This bit can be written only when H'CC is written to FRKEY in word access. 0: Issue no reset to the FCU. 1: Issues a reset to the FCU.

Note: * Write data is not retained.

26.3.13 FCU Command Register (FCMDR)

FCMDR stores the commands that the FCU has accepted. In on-chip ROM disabled mode, FCMDR is read as H'0000 and writing to it is ignored. FCMDR is initialized by a power-on reset, a transition to the hardware standby mode, or setting the FRESET bit of FRESETR to 1.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CMDR								PCMDR							
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	CMDR	H'FF	R	Command Register These bits store the latest command accepted by the FCU.
7 to 0	PCMDR	H'FF	R	Precommand Register These bits store the previous command accepted by the FCU.

Table 26.3 shows the states of FCMDR after acceptance of the various commands. For details on the blank check, see section 28.6, User Mode, User Program Mode, and User Boot Mode.

Table 26.3 FCMDR Status after a Command is Accepted

Command	CMDR	PCMDR
Normal mode transition	H'FF	Previous command
Status read mode transition	H'70	Previous command
Lock bit read mode transition (lock bit read 1)	H'71	Previous command
Program	H'E8	Previous command
Block erase	H'D0	H'20
P/E suspend	H'B0	Previous command
P/E resume	H'D0	Previous command
Status register clear	H'50	Previous command
Lock bit read 2, blank check	H'D0	H'71
Lock bit program	H'D0	H'77

26.3.14 FCU Processing Switch Register (FCPSR)

FCPSR selects a function to make the FCU suspend erasure. In on-chip ROM disabled mode, FCPSR is read as H'0000 and writing to it is ignored. FCPSR is initialized by a power-on reset, a transition to the hardware standby mode, or setting the FRESET bit of FRESETR to 1.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ESU SPMD
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	ESUSPMD	0	R/W	Erasure-Suspended Mode Selects the erasure-suspended mode to be entered when a P/E suspend command is issued while the FCU is erasing the ROM or EEPROM (see section 26.6.4, Suspending Operation). 0: Suspension-priority mode 1: Erasure-priority mode

26.3.15 Flash P/E Status Register (FPESTAT)

FPESTAT indicates the result of programming/erasure of the ROM/EEPROM. In on-chip ROM disabled mode, FPESTAT is read as H'0000 and writing to it is ignored. FPESTAT is initialized by a power-on reset, a transition to the hardware standby mode, or setting the FRESET bit of FRESETR to 1.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PEERRST							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7 to 0	PEERRST	H'00	R	P/E Error Status Indicates the source of an error that occurs during programming/erasure. This bit value is only valid if the PRGERR or ERSERR bit value in FSTATR0 is 1; otherwise the bit retains the value to indicate the source of an error that previously occurred. H'01: A write attempt made to an area protected by the lock bits H'02: A write error caused by other source than the above H'11: An erase attempt made to an area protected by the lock bits H'12: An erase error caused by other source than the above Other than above: Reserved

26.3.16 IFS Control Mode Register (IFSCMR)

The IFS control register is used to set the IFS control mode. Writing to this register in on-chip ROM disabled mode has no effect. The value read out is H'0000. This register cannot be changed during flash programming/erasure.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KEY								—	—	—	—	—	—	—	IFSCM
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R	R	R	R	R	R/W

Note: This register can be written to only when a specified value is written to the upper byte in word access. The data written to the upper byte is not stored in the register.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	KEY	All 0	R/(W)*	Key Code These bits enable or disable modification of the IFSCM bit. Data written to these bits are not stored.
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	IFSCM	0	R/W	IFS Control Mode Bit This bit should be set when setting, reset, and approval of the key code is to be performed. 0: IFS control mode is not selected. 1: IFS control mode is selected. Condition for writing to be effective: Satisfaction of all of the following conditions • On-chip ROM is in enabled mode. • FWE bit in the FPMON is 1. • FRDY bit in the FSTATR0 is 1. • H'6E is written to the KEY bits at the same time through a word-access operation

Bit	Bit Name	Initial Value	R/W	Description
0	IFSCM	0	R/W	Set condition: • Writing of 1 to IFSCM with all conditions for writing to be effective met and the value in the FENTRYR register being H'0000 Conditions for clearing: • The FRDY bit in the FSTATR0 register being 1 and the FWE bit in the FPMON is 0. • Attempting to write through byte access • Writing with a key value other than H'6E • Writing of 0 to the IFSCM register when all conditions for writing to be effective are met.

26.3.17 IFS Status Register (IFSSR)

The IFS register is used to monitor the security conditions set up by the IFS function. In on-chip ROM disabled mode, reading-out value has no effect.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	IFSST[2:0]		
Initial value:	0	0	0	0	0	0/1*	0/1*	0/1*
R/W:	R	R	R	R	R	R	R	R

Note: The initial value depends on the mode of operation and state of the key code. Refer to table 27.2.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	IFSST[2:0]	000/111*	R	IFS Status Bits These bits can be used to monitor the security conditions set up by the IFS function. 000: Unprotected mode 001: Protected–unlocked mode (security level 1) 010: Not used (not specified) 011: Protected–unlocked mode (security level 2) 100: Not used (not specified) 101: Protected–locked mode (security level 1) 110: Not used (not specified) 111: Protected–locked mode (security level 2)

Note: * The initial value depends on the mode of operation and state of the key code. Refer to table 27.2.

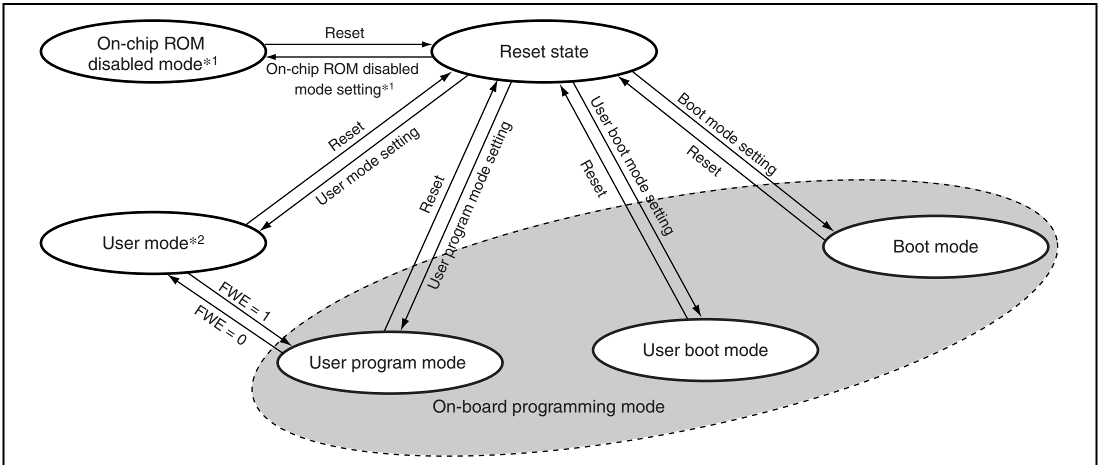
Table 26.4 IFSST [2:0] Initial Value

Operation mode	Key code setting mode	Security level for the set key code	IFSST [2:0] Initial Value	Security setting mode
ASE mode (ASEMD pin high level)	Reset	None	000	Unprotect
	Set	Level 1	101	Protect lock (Level 1)
		Level 2	—	Protect lock (Level 2) (Debug connection invalid)
Product chip mode User boot mode (ASEMD pin low level)	Reset	None	000	Unprotect
	Set	Level 1	101	Protect lock (Level 1)
		Level 2	111	Protect lock (Level 2)
Product chip mode User program mode (ASEMD pin low level)	Reset	None	111*	Protect lock (Level 2)*
	Set	Level 1	111*	Protect lock (Level 2)*
		Level 2	111*	Protect lock (Level 2)*

Note: * In the user-program mode, key-code setting conditions and the initial value of the IFSST [2:0] bits do not match. To make them match, key-code approval by the user software is required.

26.4 Overview of ROM-Related Modes

Figure 26.4 shows the ROM-related mode transition in this LSI. For the relationship between the LSI operating modes and the MD4 to MD0 and FWE pin settings, refer to section 3, Operating Modes.



Notes: 1. Modes 0 and 1 (MCU extended modes) of this LSI are on-chip ROM disabled modes.
2. Mode 2 (MCU extended mode) and mode 3 (MCU single-chip mode) of this LSI are user modes.

Figure 26.4 ROM-Related Mode Transition

- The ROM cannot be read, programmed, or erased in on-chip ROM disabled mode.
- The ROM can be read but cannot be programmed or erased in user mode.
- The ROM can be read, programmed, and erased on the board in user program mode, user boot mode, and boot mode.

Table 26.5 compares programming- and erasure-related items for the boot mode, user program mode, and user boot mode.

Table 26.5 Comparison of Programming Modes

Item	Boot Mode	User Program Mode	User Boot Mode
Programming/erasure enabled MAT	User MAT and user boot MAT	User MAT	User MAT
Programming/erasure control	Host	FCU	FCU
Programming data transfer	From host via SCI	From any device via RAM	From any device via RAM
Reset-start MAT	Embedded program stored MAT	User MAT	User boot MAT*
Transition to MCU operating mode	Mode setting change and reset	FWE setting change	Mode setting change and reset

Note: * After the LSI is started in the embedded program stored MAT and the embedded program built in the product is executed, execution starts from the location indicated by the reset vector of the user boot MAT.

- The user boot MAT can be programmed or erased only in boot mode.
- In user boot mode, a boot operation with a desired interface can be implemented through mode pin settings different from those in user program mode.
- In boot mode or user boot mode, the embedded program built in the product uses H'FFF88000 to H'FFF8FFFF in the on-chip RAM. Therefore, once the RAM is disabled via the RAM enable control register (RAMEN) and a reset is issued, the data stored in the corresponding area in the RAM prior to the reset is no longer retained in the RAM after booting is initiated in boot mode or user boot mode (see section 30, RAM).

26.5 Boot Mode

26.5.1 System Configuration

To program or erase the user MAT and user boot MAT in boot mode, send control commands and programming data from the host. The on-chip SCI of this LSI is used in asynchronous mode for communications between the host and this LSI. The tool for sending control commands and programming data must be prepared in the host. When this LSI is started in boot mode, the program in the embedded program stored MAT is executed. This program automatically adjusts the SCI bit rate and performs communications between the host and this LSI by means of the control command method.

Figure 26.5 shows the system configuration in boot mode. The NMI and $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$ interrupts are ignored in this mode, but these pins must be fixed to non-active state. Note that the AUD cannot be used in this mode.

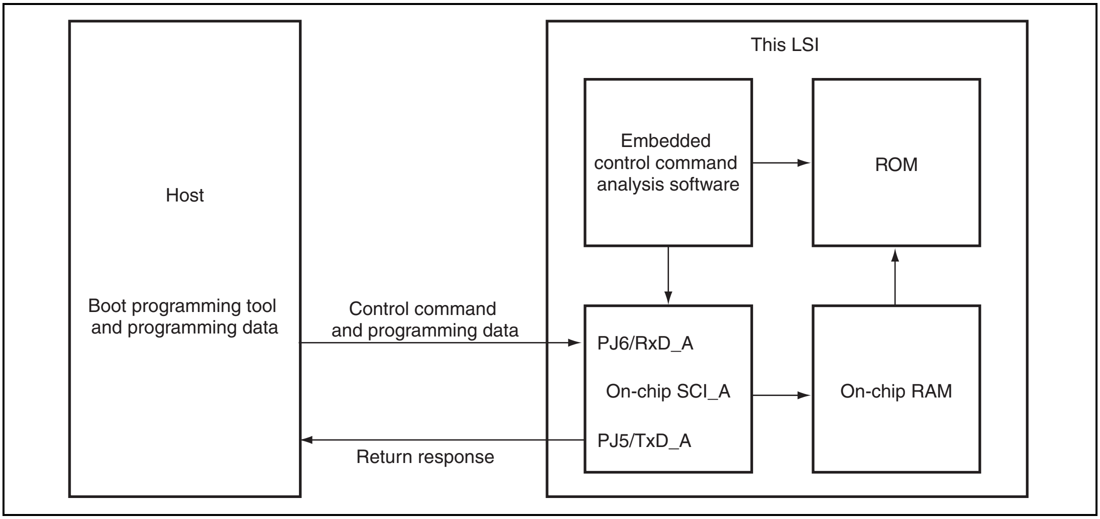

Host
Boot programming tool and programming data
Control command and programming data
Return response
This LSI
Embedded control command analysis software
ROM
PJ6/RxD_A
On-chip SCI_A
PJ5/TxD_A
On-chip RAM

Figure 26.5 System Configuration in Boot Mode

26.5.2 State Transition in Boot Mode

Figure 26.6 shows the state transition in boot mode.

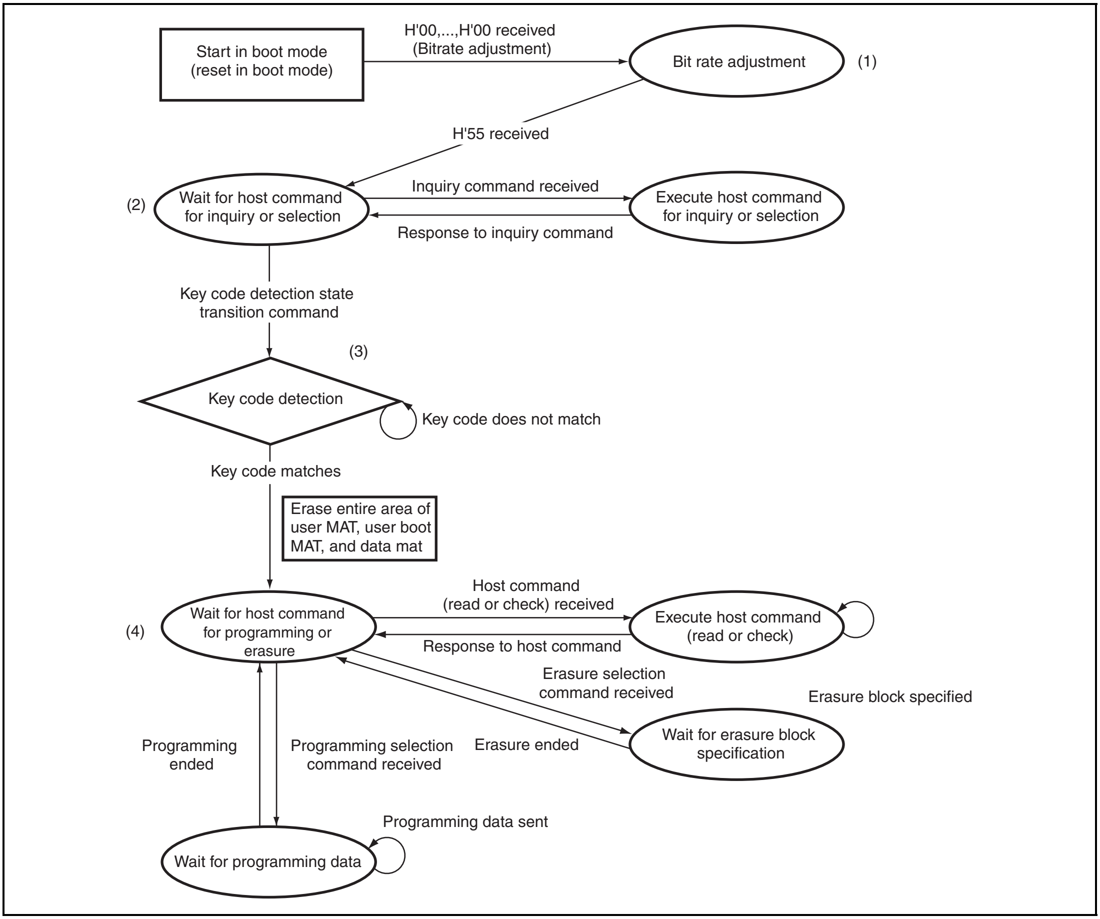


Figure 26.6 State Transition in Boot Mode

(1) Bit Rate Adjustment

After this LSI is started in boot mode, it automatically adjusts the bit rate for communications between the host and SCI_A. After automatically adjusting the bit rate, the LSI sends H'00 to the host. After the LSI has successfully received H'55 from the host, the LSI waits for a host command for inquiry or selection. For details on bit rate adjustment, see section 26.5.3, Automatic Adjustment of Bit Rate.

(2) Waiting for Host Command for Inquiry or Selection

In this state, the host inquires about the size of the MATs, structure of the MATs, addresses where the MATs start, state of support, the key code, etc., and selects the device, clock mode, and bit rate. Upon reception of a key code detection state transition command sent from the host, this LSI enters the key code detection state. For details of inquiry/selection host commands, see section 26.5.4, Inquiry/Selection Host Command Wait State.

(3) Key Code Detection

In this state, this LSI checks whether the key code sent by the host matches the one stored in this LSI. When the code matches, this LSI erases the entire area of each of the user MAT, user boot MAT, and data MAT, and enters the programming/erasure command wait state. Refer to section 27, Intelligent Flash Security (IFS), for details of key code detection.

(4) Waiting for Host Command for Programming or Erasure

In this state, this LSI performs programming or erasure in accord with commands from the host. Depending on the received command, the LSI will enter the programming data wait state, erasure block specification wait state, or command (read or check) processing state.

Upon reception of a programming selection command, the LSI waits for data to be programmed. After the programming selection command, the host sends the address where programming is to start and the data for programming. Specifying H'FFFFFFFF as the start address for programming terminates programming processing and the LSI makes a transition from the programming data wait state to the programming/erasure command wait state.

Upon reception of an erasure selection command, the LSI waits for erasure block specification. After the erasure selection command, the host sends the erasure-block number. Specifying H'FF as the erasure block number terminates erasure processing and the LSI makes a transition from the erasure block specification wait state to the programming/erasure command wait state. As the entire area of each of the user MAT, user boot MAT, and EEPROM data MAT is erased before the LSI enters the programming/erasure command wait state after it has started up in boot mode, erasure processing is not needed except in cases where data that have been programmed in boot mode must be erased without resetting the LSI.

In addition to programming and erasing commands, many other host commands are provided for use in the programming/erasure command wait state; these include commands for sum checking, blank checking (erasure checking)*, memory reading, and inquiry on the state of the user MAT and user-boot MAT. For details on these host commands, see section 26.5.5, Programming/Erasing Host Command Wait State.

Note: * Blank check function checks the erase state of the area where erase has ended. The function is disabled when programming/erasing was suspended (e.g., reset input, power-supply interruption).

26.5.3 Automatic Adjustment of Bit Rate

When this LSI is started in boot mode, it measures the low-level (H'00) period of the data that is continuously sent from the host in asynchronous SCI communications. During this measurement, set the SCI transmit/receive format to 8-bit data, 1 stop bit, and no parity, and set the bit rate to 9,600 bps or 19,200 bps. This LSI calculates the bit rate of the host SCI by means of the measured low-level period, and then sends H'00 to the host after completing the bit rate adjustment. When the host has received H'00 successfully, it must send H'55 to this LSI. If the host has failed to receive H'00, restart this LSI in boot mode to calculate and adjust the bit rate again. When this LSI has received H'55, it returns H'E6 to the host, or when it has failed to receive H'55, it returns H'FF.

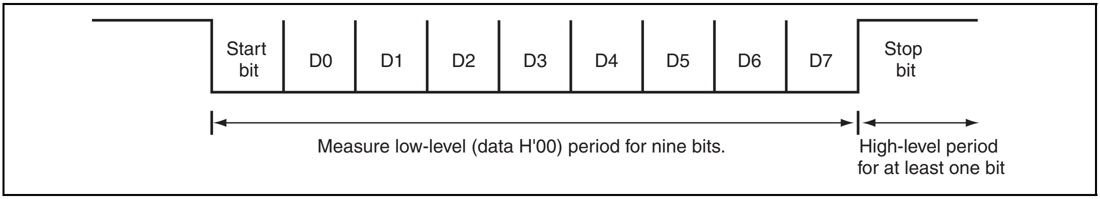

Start bit
D0
D1
D2
D3
D4
D5
D6
D7
Stop bit
Measure low-level (data H'00) period for nine bits.
High-level period for at least one bit

Figure 26.7 SCI Transmit/Receive Format for Automatic Adjustment of Bit Rate

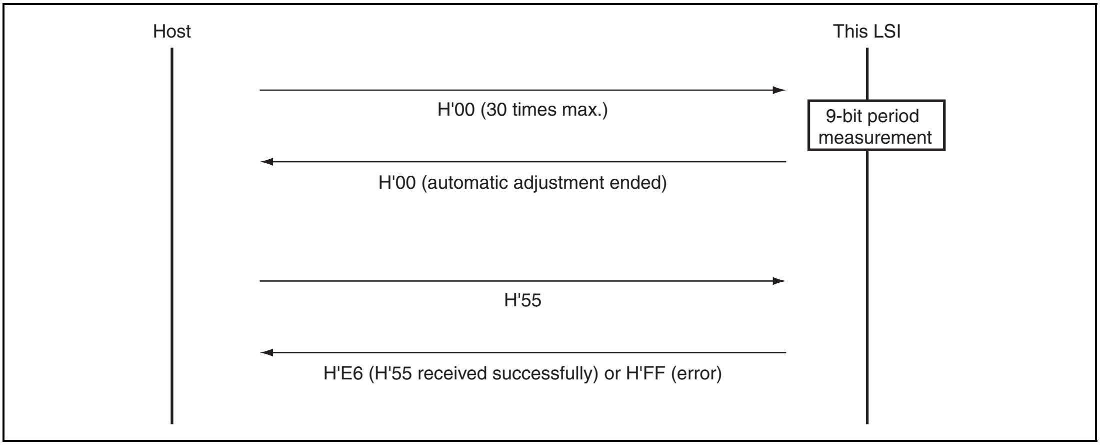

Host
This LSI
H'00 (30 times max.)
9-bit period measurement
H'00 (automatic adjustment ended)
H'55
H'E6 (H'55 received successfully) or H'FF (error)

Figure 26.8 Communication Sequence between Host and this LSI

The bit rate may not be adjusted correctly depending on the bit rate of the host SCI or the peripheral clock frequency of this LSI. Satisfy the SCI communications condition as shown in table 26.6.

Table 26.6 Condition for Automatic Adjustment of Bit Rate

Host SCI Bit Rate	Peripheral Clock Frequency of this LSI
9,600 bps	16 MHz to 20 MHz, 32 MHz to 40 MHz
19,200 bps	

26.5.4 Inquiry/Selection Host Command Wait State

Table 26.7 shows the host commands available in inquiry/selection host command wait state. The boot program status inquiry command can also be used in programming/erasure host command wait state. The key code check command can only be used in key code detection state. The other commands can only be used in inquiry/selection host command wait state.

Table 26.7 Inquiry/Selection Host Commands

Host Command Name	Function
Supported device inquiry	Inquires regarding the device codes and the product codes for the embedded programs
Device selection	Selects a device code
Clock mode inquiry	Inquires regarding the clock mode
Clock mode selection	Selects a clock mode
Multiplication ratio inquiry	Inquires regarding the number of clock types, the number of multiplication/division ratios, and the multiplication /division ratios
Operating frequency inquiry	Inquires regarding the number of clock types and the maximum and minimum operating frequencies
User boot MAT information inquiry	Inquires regarding the number of user boot MATs and the start and end addresses
User MAT information inquiry	Inquires regarding the number of user MATs and the start and end addresses
Erasure block information inquiry	Inquires regarding the number of blocks and the start and end addresses
Programming size inquiry	Inquires regarding the size of programming data
New bit rate selection	Modifies the bit rate of SCI communications between the host and this LSI
Key code detection state transition	Enters the key code detection state
Key code check command	Sends the key code
Boot program status inquiry	Inquires regarding the state of this LSI

Refer to section 27, Intelligent Flash Security (IFS), for details of key-code approval.

If the host has sent an undefined command, this LSI returns a response indicating a command error in the format shown below. The command field holds the first byte of the undefined command sent from the host.

Error response	H'80	Command

In inquiry/selection host command wait state, send selection commands from the host in the order of device selection, clock mode selection, and new bit rate selection to set up this LSI according to the responses to inquiry commands. Note that the supported device inquiry and clock mode inquiry commands are the only inquiry commands that can be sent before the clock mode selection command; other inquiry commands must not be issued before the clock mode selection command. If commands are issued in an incorrect order, this LSI returns a response indicating a command error. Figure 26.9 shows an example of the procedure to use inquiry/selection host commands.

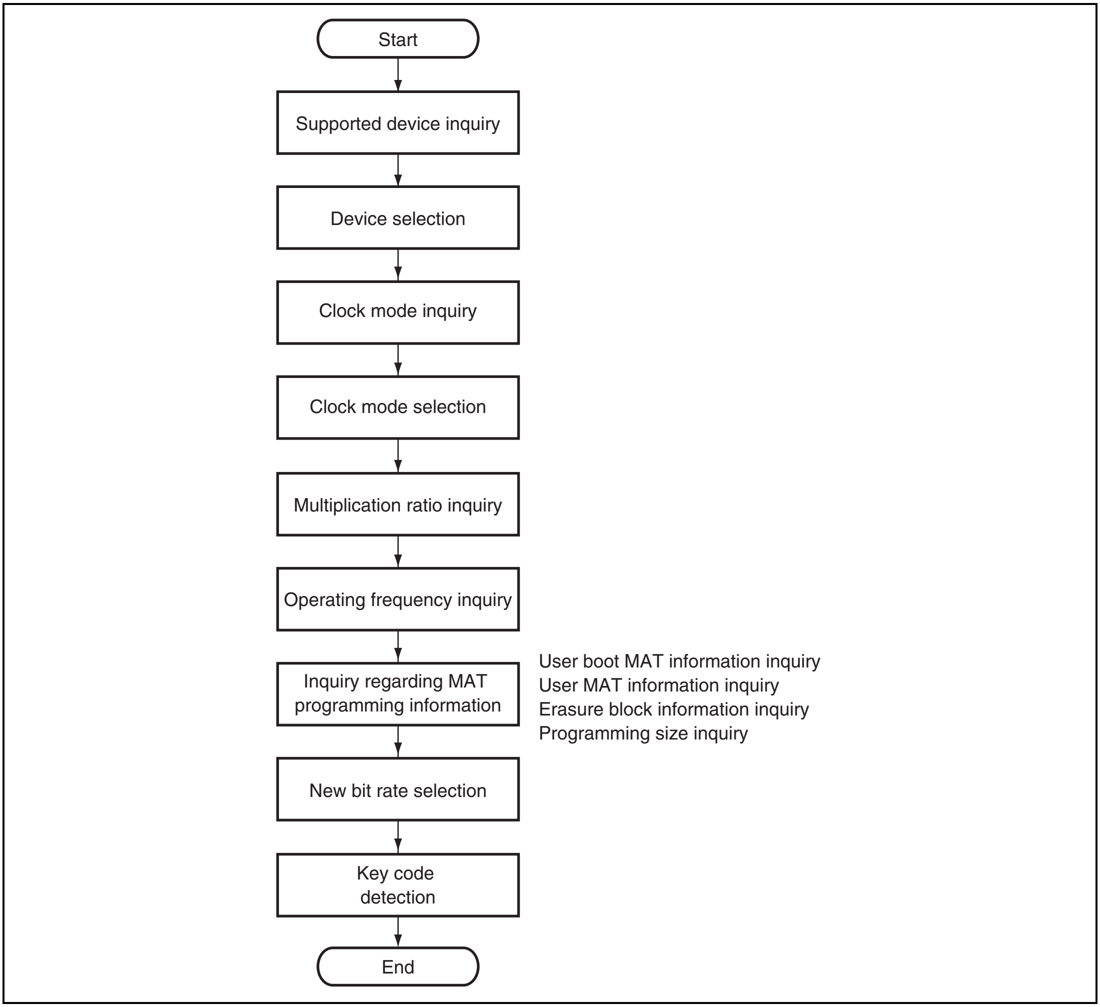

Start
Supported device inquiry
Device selection
Clock mode inquiry
Clock mode selection
Multiplication ratio inquiry
Operating frequency inquiry
Inquiry regarding MAT programming information
User boot MAT information inquiry
User MAT information inquiry
Erasure block information inquiry
Programming size inquiry
New bit rate selection
Key code detection
End

Figure 26.9 Example of Procedure to Use Inquiry/Selection Host Commands

Each host command is described in detail below. The "command" in the description indicates a command sent from the host to this LSI and the "response" indicates a response sent from this LSI to the host. The "checksum" is byte-size data calculated so that the sum of all bytes to be sent by this LSI becomes H'00.

(1) Supported Device Inquiry

In response to a supported device inquiry command sent from the host, this LSI returns the information concerning the devices supported by the embedded program for boot mode. If the supported device inquiry command comes after the host has selected a device, this LSI only returns the information concerning the selected device.

Command

H'20

Response

H'30	Size	Device count
Character count	Device code	Product code
Character count	Device code	Product code
:	:	:
Character count	Device code	Product code
SUM		

[Legend]

Size (1 byte): Total number of bytes in the device count, character count, device code, and product code fields

Device count (1 byte): Number of device types supported by the embedded program for boot mode

Character count (1 byte): Number of characters included in the device code and product code fields

Device code (4 bytes): ASCII code for the product name of the chip

Product code (n bytes): ASCII code for the name of the supported device

SUM (1 byte): Checksum

(2) Device Selection

In response to a device selection command sent from the command, this LSI checks if the selected device is supported. When the selected device is supported, this LSI specifies this device as the device for use and returns a response (H'36). If the selected device is not supported or the sent command is illegal, this LSI returns an error response (H'90).

Even when H'01 has been returned as the number of supported devices in response to a supported device inquiry command, issue a device selection command to specify the device code that has been returned as the result of the inquiry.

Command	H'10	Size	Device code	SUM
Response	H'36			
Error response	H'90	Error		

[Legend]

Size (1 byte): Number of characters in the device code field (fixed at four)

Device code (4 bytes): ASCII code for the product name of the chip (one of the device codes returned in response to the supported device inquiry command)

SUM (1 byte): Checksum

Error (1 byte): Error code

H'11: Checksum error (illegal command)

H'21: Incorrect device code error

(3) Clock Mode Inquiry

In response to a clock mode inquiry command sent from the host, this LSI returns the supported clock modes. If the clock mode inquiry command comes after the host has selected a clock mode, this LSI only returns the information concerning the selected clock mode.

Command	H'21

Response	H'31	Size		
	Mode	Mode	. . .	Mode
	SUM			

[Legend]

Size (1 byte): Total number of bytes in the mode count and mode fields

Mode (1 byte): Supported clock mode (for example, H'01 indicates clock mode 1)

SUM (1 byte): Checksum

(4) Clock Mode Selection

In response to a clock mode selection command sent from the host, this LSI checks if the selected clock mode is supported. When the selected mode is supported, this LSI specifies this clock mode for use and returns a response (H'06). If the selected mode is not supported or the sent command is illegal, this LSI returns an error response (H'91).

Be sure to issue a clock mode selection command only after issuing a device selection command. Even when H'00 or H'01 has been returned as the number of supported clock modes in response to a clock mode inquiry command, issue a clock mode selection command to specify the clock mode that has been returned as the result of the inquiry.

Command	H'11	Size	Mode	SUM
Response	H'06			
Error response	H'91	Error		

[Legend]

Size (1 byte): Number of characters in the mode field (fixed at 1)

Mode (1 byte): Clock mode (one of the clock modes returned in response to the clock mode inquiry command)

SUM (1 byte): Checksum

Error (1 byte): Error code

H'11: Checksum error (illegal command)

H'22: Incorrect clock mode error

(5) Multiplication Ratio Inquiry

In response to a multiplication ratio inquiry command sent from the host, this LSI returns the clock types, the number of multiplication/division ratios, and the multiplication division ratios supported.

Command

H'22

Response

H'32	Size	Clock type count		
Multiplication ratio count	Multiplication ratio	Multiplication ratio	. . .	Multiplication ratio
Multiplication ratio count	Multiplication ratio	Multiplication ratio	. . .	Multiplication ratio
:	:	:	. . .	:
Multiplication ratio count	Multiplication ratio	Multiplication ratio	. . .	Multiplication ratio
SUM				

[Legend]

Size (1 byte): Total number of bytes in the clock type count, multiplication ratio count, and multiplication ratio fields

Clock type count (1 byte): Number of clock types (for example, H'02 indicates two clock types; that is, an internal clock and a peripheral clock)

Multiplication ratio count (1 byte): Number of supported multiplication/division ratios (for example, H'03 indicates that three multiplication ratios are supported for the internal clock (x4, x6, and x8))

Multiplication ratio (1 byte): A positive value indicates a multiplication ratio (for example, H'04 = 4 = multiplication by 4)

A negative value indicates a division ratio (for example, H'FE = -2 = division by 2)

SUM (1 byte): Checksum

(6) Operating Clock Frequency Inquiry

In response to an operating clock frequency inquiry command sent from the host, this LSI returns the minimum and maximum frequencies for each clock.

Command

H'23

Response

H'33	Size	Clock type count	
Minimum frequency		Maximum frequency	
Minimum frequency		Maximum frequency	
:		:	
Minimum frequency		Maximum frequency	
SUM			

[Legend]

Size (1 byte): Total number of bytes in the clock type count, minimum frequency, and maximum frequency fields

Clock type count (1 byte): Number of clock types (for example, H'02 indicates two clock types; that is, an internal clock and a peripheral clock)

Minimum frequency (2 bytes): Minimum value of the operating frequency (for example, H'07D0 indicates 20.00 MHz).

This value should be calculated by multiplying the frequency value (MHz) to two decimal places by 100.

Maximum frequency (2 bytes): Maximum value of the operating frequency represented in the same format as the minimum frequency

SUM (1 byte): Checksum

(7) User Boot MAT Information Inquiry

In response to a user boot MAT information inquiry command sent from the host, this LSI returns the number of user boot MATs and their addresses.

Command

H'24		

Response

H'34	Size	MAT count
MAT start address		
MAT end address		
MAT start address		
MAT end address		
:		
MAT start address		
MAT end address		
SUM		

[Legend]

Size (1 byte): Total number of bytes in the MAT count, MAT start address, and MAT end address fields

MAT count (1 byte): Number of user boot MATs (consecutive areas are counted as one MAT)

MAT start address (4 bytes): Start address of a user boot MAT

MAT end address (4 bytes): End address of a user boot MAT

SUM (1 byte): Checksum

(8) User MAT Information Inquiry

In response to a user MAT information inquiry command sent from the host, this LSI returns the number of user MATs and their addresses.

Command	H'25

Response	H'35	Size	MAT count
	MAT start address		
	MAT end address		
	MAT start address		
	MAT end address		
	:		
	MAT start address		
	MAT end address		
	SUM		

[Legend]

Size (1 byte): Total number of bytes in the MAT count, MAT start address, and MAT end address fields

MAT count (1 byte): Number of user MATs (consecutive areas are counted as one MAT)

MAT start address (4 bytes): Start address of a user MAT

MAT end address (4 bytes): End address of a user MAT

SUM (1 byte): Checksum

(9) Erasure Block Information Inquiry

In response to an erasure block information inquiry command sent from the host, this LSI returns the number of erasure blocks in the user MAT and their addresses.

Command

H'26

Response

<table>
<tr><td>H'36</td><td>Size</td><td>Block count</td></tr>
<tr><td colspan="3">Block start address</td></tr>
<tr><td colspan="3">Block end address</td></tr>
<tr><td colspan="3">Block start address</td></tr>
<tr><td colspan="3">Block end address</td></tr>
<tr><td colspan="3">:</td></tr>
<tr><td colspan="3">Block start address</td></tr>
<tr><td colspan="3">Block end address</td></tr>
<tr><td>SUM</td><td></td><td></td></tr>
</table>

[Legend]

Size (2 bytes): Total number of bytes in the block count, block start address, and block end address fields

Block count (1 byte): Number of erasure blocks in the user MAT

Block start address (4 bytes): Start address of an erasure block

Block end address (4 bytes): End address of an erasure block

SUM (1 byte): Checksum

(10) Programming Size Inquiry

In response to a programming size inquiry command sent from the host, this LSI returns the programming size.

Command	H'27			
Response	H'37	Size	Programming size	SUM

[Legend]

Size (1 byte): Number of characters included in the programming size field (fixed at two)

Programming size (2 bytes): Programming unit (bytes)

SUM (1 byte): Checksum

(11) New Bit Rate Selection

In response to a new bit rate selection command sent from the host, this LSI checks if the on-chip SCI can be set to the selected new bit rate. When the SCI can be set to the new bit rate , this LSI returns a response (H'06) and sets the SCI to the new bit rate. If the SCI cannot be set to the new bit rate or the sent command is illegal, this LSI returns an error response (H'BF). Upon reception of response H'06, the host waits for a one-bit period in the previous bit rate with which the new bit rate selection command has been sent, and then sets the host bit rate to the new one. After that, the host sends confirmation data (H'06) in the new bit rate, and this LSI returns a response (H'06) to the confirmation data.

Be sure to issue a new bit rate selection command only after a clock mode selection command.

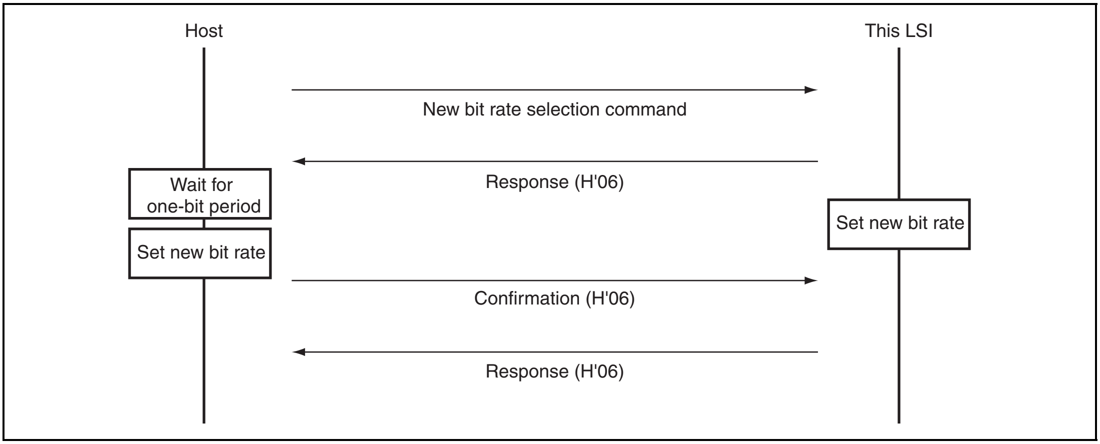


Figure 26.10 New Bit Rate Selection Sequence

Command	H'3F	Size	Bit rate	Input frequency
	Clock type count	Multiplication ratio 1	Multiplication ratio 2	
	SUM			

Response	H'06	
Error response	H'BF	Error
Confirmation	H'06	
Response	H'06	

[Legend]

Size (1 byte): Total number of bytes in the bit rate, input frequency, clock type count, and multiplication ratio fields

Bit rate (2 bytes): New bit rate (for example, H'00C0 indicates 19200 bps)

1/100 of the new bit rate value should be specified.

Input frequency (2 bytes): Clock frequency input to this LSI (for example, H'07D0 indicates 20.00 MHz)

This value should be calculated by multiplying the input frequency value to two decimal places by 100.

Clock type count (1 byte): Number of clock types (for example, H'02 indicates two clock types; that is, an internal clock and a peripheral clock)

Multiplication ratio 1 (1 byte): Multiplication/division ratio of the input frequency to obtain the internal clock

A positive value indicates a multiplication ratio (for example, H'04 = 4 = multiplication by 4)

A negative value indicates a division ratio (for example, HFE = -2 = division by 2)

Multiplication 2 (1 byte): Multiplication/division ratio of the input frequency to obtain the peripheral clock

This value is represented in the same format as multiplication ratio 1

SUM (1 byte): Checksum

Error: Error code

H'11: Checksum error

H'24: Bit rate selection error

H'25: Input frequency error

H'26: Multiplication ratio error

H'27: Operating frequency error

- Bit rate selection error

  A bit rate selection error occurs when the bit rate selected through a new bit rate selection command cannot be set for the SCI of this LSI within an error of 4%. The bit rate error can be obtained by the following equation from the bit rate (B) selected through a new bit rate selection command, the input frequency (fEX), multiplication ratio 2 (MPϕ), the SCBRR1A setting (N) in SCI_A, and the CKS[1:0] bit value (n) in SCSMR1A.

$$\text{Error (\%)} = \left\{ \frac{f_{EX} \times M_{P\phi} \times 10^6}{(N+1) \times B \times 32 \times 2^{2n-1}} - 1 \right\} \times 100$$

- Input frequency error

  An input frequency error occurs when the input frequency specified through a new bit rate selection command is outside the range from the minimum to maximum input frequencies for the clock mode selected through a clock mode selection command.

- Multiplication ratio error

  A multiplication ratio error occurs when the multiplication ratio specified through a new bit rate selection command does not match the clock mode selected through a clock mode selection command. To check the selectable multiplication ratios, issue a multiplication ratio inquiry command.

- Operating frequency error

  An operating frequency error occurs when this LSI cannot operate at the operating frequencies selected through a new bit rate selection command. This LSI calculates the operating frequencies from the input frequency and multiplication ratios specified through a new bit rate selection command and checks if each calculated frequency is within the range from the minimum to maximum frequencies for the respective clock. To check the minimum and maximum operating frequencies for each clock, issue an operating clock frequency inquiry command.

(12) Key Code Detection State Transition

When receiving a key code detection state transition command, this LSI enters the key code detection state.

Command	H'40
Response	H'16

(13) Key Code Check Command

On reception of a key code check command, this LSI performs the approval with the key code sent from the host. After successful verification of the key code, this LSI erases the entire area of each of the user MAT, user boot MAT, and data MAT and enters the programming/erasure host command wait state. If this LSI fails to verify the key code due to the key code being in error, it returns an error response (sends H'E0 and H'61 in that order).

Command	H'60	Total number of bytes	Key code (16 bytes)	SUM
Response	H'26			
Error response	H'E0	Error		

[Legend]

Total number of bytes (1 byte): Total number of bytes in the status and error fields; fixed at H'10 (16 bytes)

Key code (16 bytes): Key-code setting; pad higher-order bits that are not required with H'FF.

SUM (1 byte): Checksum; summary from command to SUM, set as H'00.

Error (1 byte): Error state of this LSI (see table 26.9)

(14) Boot Program Status Inquiry

In response to a boot program status inquiry command sent from the host, this LSI returns its current status. The boot program status inquiry command can be issued in both inquiry/selection host command wait state and programming/erasure host command wait state.

Command	H'4F

Response	H'5F	Size	Status	Error	SUM

[Legend]

Size (1 byte): Total number of bytes in the status and error fields (fixed at two)

Status (1 byte): Current status in this LSI (see table 26.8)

Error (1 byte): Error status in this LSI (see table 26.9)

SUM (1 byte): Checksum

Table 26.8 Status Code

Code	Description
H'11	Waiting for device selection
H'12	Waiting for clock mode selection
H'13	Waiting for new bit rate selection
H'1F	Waiting for transition to key code detection state (new bit rate has been selected)
H'3F	Waiting for a programming/erasure host command
H'4F	Waiting for reception of programming data
H'5F	Waiting for erasure block selection

Table 26.9 Error Code

Code	Description
H'00	No error
H'11	Checksum error
H'21	Incorrect device code error
H'22	Incorrect clock mode error
H'24	Bit rate selection error
H'25	Input frequency error
H'26	Multiplication ratio error
H'27	Operating frequency error
H'29	Block number error
H'2A	Address error
H'2B	Data size error
H'51	Erasure error
H'52	Incomplete erasure error
H'53	Programming error
H'61	Incorrect key code error
H'80	Command error
H'FF	Bit rate adjustment verification error

26.5.5 Programming/Erasing Host Command Wait State

Table 26.10 shows the host commands available in programming/erasure host command wait state.

Table 26.10 Programming/Erasure Host Commands

Host Command Name	Function
User boot MAT programming selection	Selects the program for user boot MAT programming
User MAT programming selection	Selects the program for user MAT programming
256-byte programming	Programs 256 bytes of data
Erasure selection	Selects the erasure program
Block erasure	Erases block data
Memory read	Reads data from memory
User boot MAT checksum	Performs checksum verification for the user boot MAT
User MAT checksum	Performs checksum verification for the user MAT
User boot MAT blank check	Checks whether the user boot MAT is blank
User MAT blank check	Checks whether the user MAT is blank
Read lock bit status	Reads a lock bit
Lock bit program	Programs a lock bit
Enable lock bit	Enables lock-bit protection
Disable lock bit	Disables lock-bit protection
Boot program status inquiry	Inquires regarding the state of this LSI
Key code set	Sets the key code
Key code reset	Resets the key code

If the host has sent an undefined command, this LSI returns a response indicating a command error. For the format of this response, see section 26.5.4, Inquiry/Selection Host Command Wait State.

To program the ROM, issue a programming selection command (user boot MAT programming selection or user MAT programming selection command) and then a 256-byte programming command from the host. Upon reception of a programming selection command, this LSI enters programming data wait state (see section 26.5.2, State Transition in Boot Mode). In response to a 256-byte programming command sent from the host in this state, this LSI starts programming the ROM. When the host sends a 256-byte programming command specifying H'FFFFFFFF as the programming start address, this LSI detects it as the end of programming and enters programming/erasure host command wait state.

To erase the ROM, issue an erasure selection command and then a block erasure command from the host. Upon reception of an erasure selection command, this LSI enters erasure block selection wait state (see section 26.5.2, State Transition in Boot Mode). In response to a block erasure command sent from the host in this state, this LSI erases the specified block in the ROM. When the host sends a block erasure command specifying H'FF as the block number, this LSI detects it as the end of erasure and enters programming/erasure host command wait state.

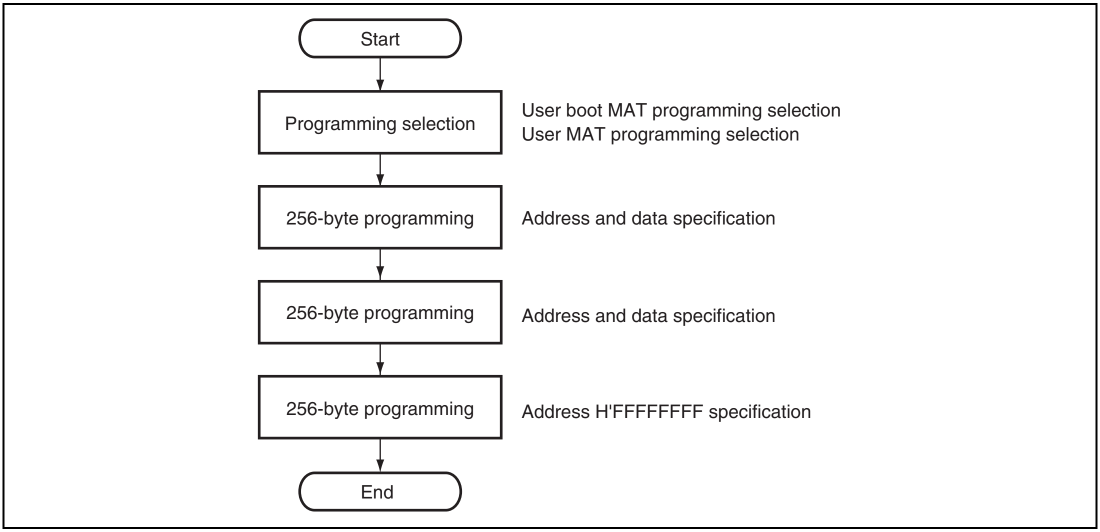

Start
Programming selection
User boot MAT programming selection
User MAT programming selection
256-byte programming
Address and data specification
256-byte programming
Address and data specification
256-byte programming
Address H'FFFFFFFF specification
End

Figure 26.11 Procedure for ROM Programming in Boot Mode

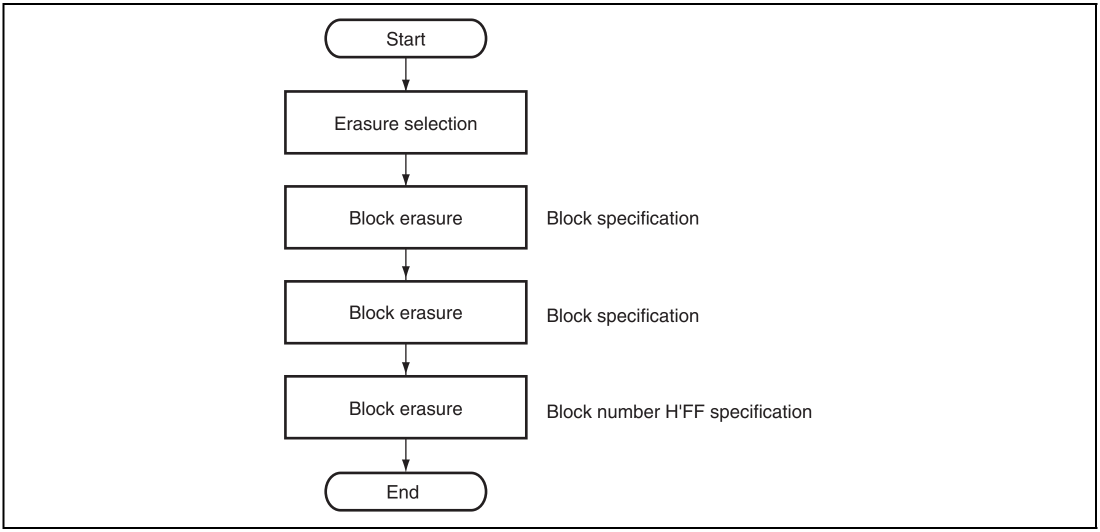


Figure 26.12 Procedure for ROM Erasure in Boot Mode

Each host command is described in detail below. The "command" in the description indicates a command sent from the host to this LSI and the "response" indicates a response sent from this LSI to the host. The "checksum" is byte-size data calculated so that the sum of all bytes to be sent by this LSI becomes H'00.

(1) User Boot MAT Programming Selection

In response to a user boot MAT programming selection command sent from the host, this LSI selects the program for user boot MAT programming and waits for programming data.

Command	H'42
Response	H'06

(2) User MAT Programming Selection

In response to a user MAT programming selection command sent from the host, this LSI selects the program for user MAT programming and waits for programming data.

Command	H'43
Response	H'06

(3) 256-Byte Programming

In response to a 256-byte programming command sent from the host, this LSI programs the ROM. After completing ROM programming successfully, this LSI returns a response (H'06). If an error has occurred during ROM programming, this LSI returns an error response (H'D0).

Command	H'50	Programming Address		
	Data	Data	. . .	Data
	SUM			

Response	H'06

Error response	H'D0	Error

[Legend]

Programming address (4 bytes): Target address of programming

To program the ROM, a 256-byte boundary address should be specified.

To terminate programming, H'FFFFFFFF should be specified.

Data (256 bytes): Programming data

H'FF should be specified for the bytes that do not need to be programmed.

When terminating programming, no data needs to be specified (only the programming address and SUM should be sent in that order).

SUM (1 byte): Checksum

Error (1 byte): Error code

H'11: Checksum error

H'2A: Address error (the specified address is not in the target MAT)

H'53: Programming cannot be done due to a programming error

(4) Erasure Selection

In response to an erasure selection command sent from the host, this LSI selects the erasure program and waits for erasure block specification.

Command	H'48
Response	H'06

(5) Block Erasure

In response to a block erasure command sent from the host, this LSI erases the ROM. After completing ROM erasure successfully, this LSI returns a response (H'06). If an error has occurred during ROM erasure, this LSI returns an error response (H'D8).

Command	H'58	Size	Block	SUM
Response	H'06			
Error response	H'D8	Error		

[Legend]

Size (1 byte): Number of bytes in the block specification field (fixed at 1)

Block (1 byte): Block number whose data is to be erased
To terminate erasure, H'FF should be specified.

SUM (1 byte): Checksum

Error (1 byte): Error code
H'11: Checksum error
H'29: Block number error (an incorrect block number is specified)
H'51: Erasure cannot be done due to an erasure error

(6) Memory Read

In response to a memory read command sent from the host, this LSI reads data from the ROM. After completing ROM reading, this LSI returns the data stored in the address specified by the memory read command. If this LSI has failed to read the ROM, this LSI returns an error response (H'D2).

Command

H'52	Size	Area	Read start address
Reading size			SUM

Response

H'52	Reading size		
Data	Data	. . .	Data
SUM			

Error response

H'D2	Error

[Legend]

Size (1 byte): Total number of bytes in the area, read start address, and reading size fields

Area (1 byte): Target MAT to be read

H'00: User boot MAT

H'01: User MAT

Read start address (4 bytes): Start address of the area to be read

Reading size (4 bytes): Size of data to be read (bytes)

SUM (1 byte): Checksum

Data (1 byte): Data read from the ROM

Error (1 byte): Error code

H'11: Checksum error

H'2A: Address error

- The value specified for area selection is neither H'00 nor H'01.
- The specified read start address is outside the selected MAT.

H'2B: Data size error

- H'00 is specified for the reading size.
- The reading size is larger than the MAT.
- The end address calculated from the read start address and the reading size is outside the selected MAT.

(7) User Boot MAT Checksum

In response to a user boot MAT checksum command sent from the host, this LSI sums the user boot MAT data in byte units and returns the result (checksum).

Command	H'4A			
Response	H'5A	Size	MAT checksum	SUM

[Legend]

Size (1 byte): Number of bytes in the MAT checksum field (fixed at 4)

MAT checksum (4 bytes): Checksum of the user boot MAT data

SUM (1 byte): Checksum (for the response data)

(8) User MAT Checksum

In response to a user MAT checksum command sent from the host, this LSI sums the user MAT data in byte units and returns the result (checksum).

Command	H'4B			
Response	H'5B	Size	MAT checksum	SUM

[Legend]

Size (1 byte): Number of bytes in the MAT checksum field (fixed at 4)

MAT checksum (4 bytes): Checksum of the user MAT data

SUM (1 byte): Checksum (for the response data)

(9) User Boot MAT Blank Check

In response to a user boot MAT blank check command sent from the host, this LSI checks whether the user boot MAT is completely erased. When the user boot MAT is completely erased, this LSI returns a response (H'06). If the user boot MAT has an unerased area, this LSI returns an error response (sends H'CC and H'52 in that order).

Command	H'4C	
Response	H'06	
Error response	H'CC	H'52

No verification function is provided to check program/erase state of the area where the data is undefined by suspend of program/erase (e.g., reset input, power-supply interruption). Therefore, if the undefined area should be used again, make sure to completely erase data before usage.

(10) User MAT Blank Check

In response to a user MAT blank check command sent from the host, this LSI checks whether the user MAT is completely erased. When the user MAT is completely erased, this LSI returns a response (H'06). If the user MAT has an unerased area, this LSI returns an error response (sends H'CD and H'52 in that order).

Command	H'4D	
Response	H'06	
Error response	H'CD	H'52

No verification function is provided to check program/erase state of the area where the data is undefined by suspend of program/erase (e.g., reset input, power-supply interruption). Therefore, if the undefined area should be used again, make sure to completely erase data before usage.

(11) Read Lock Bit Status

In response to a read lock bit status command sent from the host, this LSI reads data from the lock bit. After completing the lock bit reading, this LSI returns the data stored in the address specified by the read lock bit status command. If this LSI has failed to read the lock bit, this LSI returns an error response (H'F1).

Command	H'71	Size	Area	Medium address	Upper address	SUM

Response	Status

Error response	H'F1	Error

[Legend]

Size (1 byte): Total number of bytes in the area, medium address, and upper address (fixed at 3 in this LSI)

Area (1 byte): Target MAT to be read

H'00: User boot MAT

H'01: User MAT

Medium address (1 byte): Medium address at the end of the specified address (8 to 15 bits)

Upper address (1 byte): Upper address at the end of the specified address (16 to 23 bits)

SUM (1 byte): Checksum

Status (1 byte): Bit 6 locked at "0"

: Bit 6 unlocked at "1"

Error (1 byte): Error code

H'11: Checksum error

H'2A: Address error (the specified address is not in the target MAT)

(12) Lock Bit Program

In response to a lock bit program command sent from the host, this LSI writes to a lock bit and locks the specified block. After completing the lock bit blocking, this LSI returns a response (H'06). If this LSI has failed to lock, this LSI returns an error response (H'F7).

Command	H'77	Size	Area	Medium address	Upper address	SUM

Response	H'16

Error response	H'F7	Error

[Legend]

Size (1 byte): Total number of bytes in the area, medium address, and upper address (fixed at 3 in this LSI)

Area (1 byte): Target MAT to be locked

H'00: User boot MAT

H'01: User MAT

Medium address (1 byte): Medium address at the end of the specified address (8 to 15 bits)

Upper address (1 byte): Upper address at the end of the specified address (16 to 23 bits)

SUM (1 byte): Checksum

Error (1 byte): Error code

H'11: Checksum error

H'2A: Address error (the specified address is not in the target MAT)

H'53: Locking cannot be done due to a programming error

(13) Lock Bit Enable

In response to a lock bit enable command from the host, this enables the lock bits.

Command	H'7A
Response	H'06

(14) Lock Bit Disable

In response to a lock bit disable command from the host, this LSI disables the lock bits.

Command	H'75
Response	H'06

(15) Boot Program Status Inquiry

For details, refer to section 26.5.4, Inquiry/Selection Host Command Wait State.

(16) Key Code Set Command

In response to a key code set command from the host, this LSI sets the key code. When the key code set is completed successfully, this LSI returns a response (H'06). If an error has occurred during the key code set processing, this LSI returns an error response (H'D4).

Command	H'54	Number of data bytes	Security level	Key code	SUM

Response	H'06

Error response	H'D4	Error

[Legend]

Number of data bytes (1 byte): The number of bytes in the security level and the key code (fixed at H'11 in this LSI)

Security level (1 byte): Setting the security level of the key code to be set

H'01: Security level 1

H'02: Security level 2

If neither H'01 nor H'02 is set, a security level error will occur.

Key code (16 bytes): Key code

For the higher-order 8 bytes, input H'FF. For the lower-order 8 bytes, input the key code (in 8 bytes).

SUM (1 byte): Checksum

Set the checksum from the command to the SUM to be H'00.

Error (1 byte): Error code

H'11: Checksum error

H'53: Key code set error

H'64: Security level error

(17) Key Code Reset Command

In response to a key code reset command form the host, this LSI resets the key code. When the key code reset is completed successfully, this LSI returns a response (H'06). If an error has occurred during the key code reset processing, this LSI returns an error response (H'DC).

Command	H'5C	Number of data bytes	Key code area	SUM

Response	H'06

Error response	H'DC	Error

[Legend]

Number of data bytes (1 byte): The number of bytes in the key code area
(fixed at H'01 in this LSI)

Key code area (1 byte): Specifies the key code area
(fixed at H'00 in this LSI)

SUM (1 byte): Checksum
Set the checksum from the command to the SUM to be H'00.

Error (1 byte): Error code
H'11: Checksum error
H'51: Key code reset error

26.6 User Program Mode

26.6.1 FCU Command List

To program or erase the user MAT in user program mode, issue FCU commands to the FCU. Table 26.11 is a list of FCU commands for ROM programming and erasure. For details of the FCU commands related to the intelligent flash security, refer to section 27.3.4, Key Code Set, Reset, and Approval.

Table 26.11 FCU Command List (ROM-Related Commands)

Command	Function
Normal mode transition	Moves to the normal mode (see section 26.6.2, Conditions for FCU Command Acceptance)
Status read mode transition	Moves to the status read mode (see section 26.6.2, Conditions for FCU Command Acceptance)
Lock bit read mode transition (lock bit read 1)	Moves to the lock bit read mode (see section 26.6.2, Conditions for FCU Command Acceptance)
Program	Programs ROM (in 256-byte units)
Block erase	Erases ROM (in block units; erasing the lock bit)
P/E suspend	Suspends programming or erasure
P/E resume	Resumes programming or erasure
Status register clear	Clears the ILGLERR, ERSERR, and PRGERR bits in FSTATR0 and cancels the command-locked state
Lock bit read 2	Reads the lock bit of a specified erasure block (updates the FLOCKST bit in FSTATR1 to reflect the lock bit state)
Lock bit program	Writes to the lock bit of a specified erasure block

FCU commands other than the lock bit read 2 program and lock bit program are also used for EEPROM programming and erasure. When a lock bit read 2 command is issued to the EEPROM, an EEPROM blank check is executed. When a lock bit program command is issued to the EEPROM, it is detected as an illegal command and generates an error (see section 28, EEPROM).

To issue a command to the FCU, write to a ROM program/erase address through the P bus. Table 26.12 shows the FCU command format. Performing P-bus write access as shown in table 26.11 under specified conditions starts each command processing in the FCU. For the conditions for FCU command acceptance, refer to section 26.6.2, Conditions for FCU Command Acceptance. For details of each FCU command, refer to section 26.6.3, FCU Command Usage.

When H'71 is sent in the first cycle of an FCU command while the FRDMD bit is 0 (memory area read mode), the FCU accepts the lock bit read mode transition command (lock bit read 1). When a ROM program/erase address is read through the P bus after transition to the lock bit read mode, the FCU copies the lock bit of the erasure block corresponding to the accessed address into all bits in the read data. When H'71 is sent in the first cycle of the FCU command while the FRDMD bit is 1 (register read mode), the FCU waits for the second-cycle data (H'D0) of the lock bit read 2 command. When a ROM program/erase address is written to through the P bus in this state, the FCU copies the lock bit of the erasure block corresponding to the accessed address into the FLOCKST bit in FSTATR1.

There are two suspending modes to be initiated by the P/E suspend command; the suspension-priority mode and erasure-priority mode. For details of each mode, refer to section 26.6.4, Suspending Operation.

Table 26.12 FCU Command Format

	Number of Bus	First Cycle		Second Cycle		Third Cycle		Fourth to 130th Cycles		131st Cycle	
Command	Cycles	Address	Data	Address	Data	Address	Data	Address	Data	Address	Data
Normal mode transition	1	RA	H'FF	—	—	—	—	—	—	—	—
Status read mode transition	1	RA	H'70	—	—	—	—	—	—	—	—
Lock bit read mode transition (lock bit read 1)	1	RA	H'71	—	—	—	—	—	—	—	—
Program	131	RA	H'E8	RA	H'80	WA	WD1	RA	WDn	RA	H'D0
Block erase	2	RA	H'20	BA	H'D0	—	—	—	—	—	—
P/E suspend	1	RA	H'B0	—	—	—	—	—	—	—	—
P/E resume	1	RA	H'D0	—	—	—	—	—	—	—	—
Status register clear	1	RA	H'50	—	—	—	—	—	—	—	—
Lock bit read 2	2	RA	H'71	BA	H'D0	—	—	—	—	—	—
Lock bit program	2	RA	H'77	BA	H'D0	—	—	—	—	—	—

[Legend]

RA: ROM program/erase address
When FENTRY0 is 1: An address in the range from H'80800000 to H'808FFFFF
When FENTRY1 is 1: An address in the range from H'80900000 to H'809FFFFF
When FENTRY2 is 1: An address in the range from H'80A00000 to H'80AFFFFF
When FENTRY3 is 1: An address in the range from H'80B00000 to H'80B3FFFF
When FENTRY4 is 1: An address in the range from H'80B40000 to H'80B7FFFF
When FENTRY5 is 1: An address in the range from H'80B80000 to H'80BBFFFF
When FENTRY6 is 1: An address in the range from H'80BC0000 to H'80BFFFFF

WA: ROM program address
Start address of 256-byte programming data

BA: ROM erasure block address
An address in the target erasure block (specified by the ROM program/erase address)

WDn: n-th word of programming data (n = 1 to 128)

26.6.2 Conditions for FCU Command Acceptance

The FCU determines whether to accept a command depending on the FCU mode or status. Figure 26.13 is an FCU mode transition diagram.

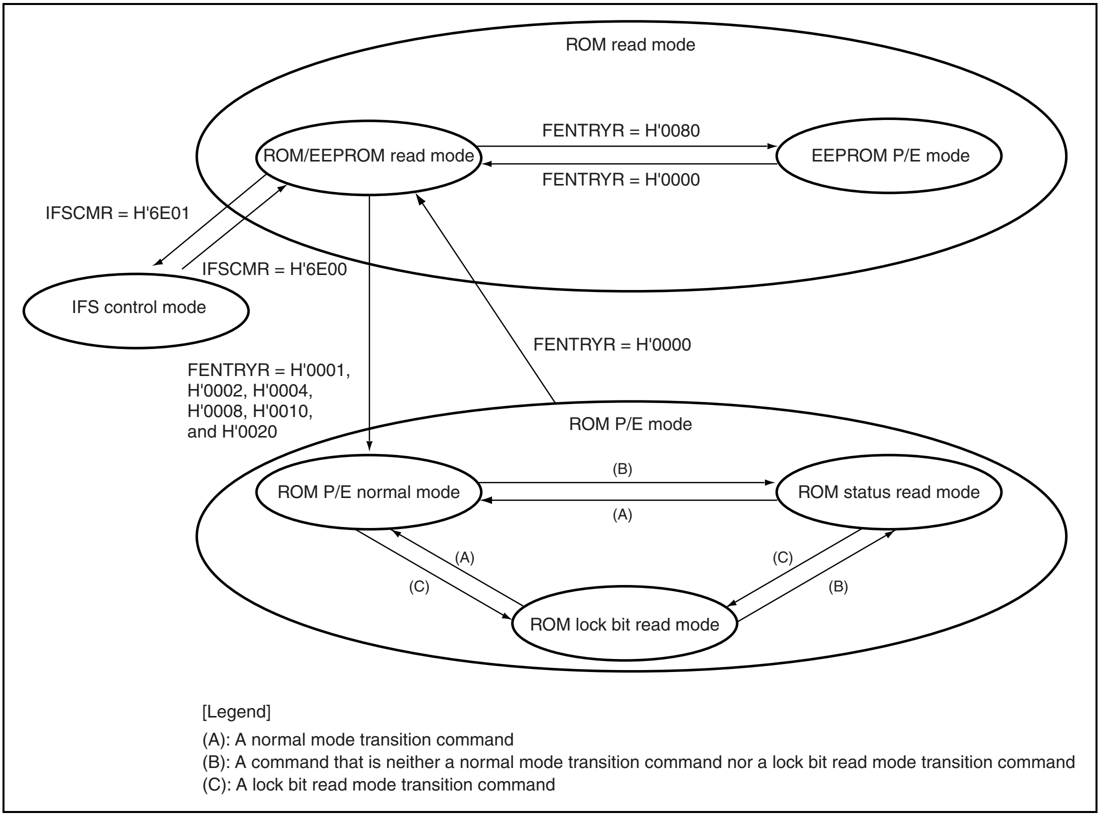

ROM read mode
ROM/EEPROM read mode
FENTRYR = H'0080
FENTRYR = H'0000
EEPROM P/E mode
IFSCMR = H'6E01
IFSCMR = H'6E00
IFS control mode
FENTRYR = H'0000
FENTRYR = H'0001, H'0002, H'0004, H'0008, H'0010, and H'0020
ROM P/E mode
ROM P/E normal mode
(B)
(A)
ROM status read mode
(A)
(C)
(C)
(B)
ROM lock bit read mode
[Legend]
(A): A normal mode transition command
(B): A command that is neither a normal mode transition command nor a lock bit read mode transition command
(C): A lock bit read mode transition command

Figure 26.13 FCU Mode Transition Diagram (ROM-Related Modes)

For details of the FCU command acceptance in IFS control mode, refer to section 27.3.4, Key Code Set, Reset, and Approval.

(1) ROM Read Mode

- ROM/EEPROM read mode

  The ROM and EEPROM can be read through the ROM cache and peripheral bus, respectively, at a high speed. The FCU does not accept commands. The FCU enters this mode when the FENTRY6 to FENTRY0 bits in FENTRYR are set to 0000000 and the FENTRYD bit to 0 in FENTRYR.

  When transiting to ROM read mode by changing FENTRY6 to FENTRY0 bit in FENTRYR register from 1 to 0, write 0 to FENTRY6 to FENTRY0 bit and execute a minimum of 5 NOP instructions after performing a dummy read of FENTRYR register.
- EEPROM P/E mode

  The ROM can be read through the ROM cache at a high speed. The FCU accepts commands for EEPROM, but does not accept commands for ROM. The FCU enters this mode when the FENTRY6 to FENTRY0 bits are set to 0000000 and the FENTRYD bit to 1. For details of the EEPROM P/E mode, refer to section 28.6.2, Conditions for FCU Command Acceptance.

(2) ROM P/E Mode

- ROM P/E normal mode

  The FCU enters this mode when the FENTRYD bit is set to 0 and any one of the bits, FENTRY6 to FENTRY0, is set to 1 in ROM read mode, or when a normal mode transition command is accepted in ROM P/E mode. Table 26.13 shows the commands that can be accepted in this mode. High-speed read operation is not available for the ROM. If an address in the range from H'80BC0000 to H'80BFFFFF is read through the P-bus while the FENTRY6 bit is set to 1, an address in the range from H'80B80000 to H'80BBFFFF is read through the P-bus while the FENTRY5 bit is set to 1, an address in the range from H'80B40000 to H'80B7FFFF is read through the P-bus while the FENTRY4 bit is set to 1, an address in the range from H'80B00000 to H'80B3FFFF is read through the P-bus while the FENTRY3 bit is set to 1, an address in the range from H'80A00000 to H'80AFFFFF is read through the P-bus while the FENTRY2 bit is set to 1, an address in the range from H'80900000 to H'809FFFFF is read through the P-bus while the FENTRY1 bit is set to 1, or an address in the range from H'80800000 to H'808FFFFF is read through the P-bus while the FENTRY0 bit is set to 1, a ROM access error occurs and the FCU enters the command-locked state (see section 26.8.3, Error Protection).

- ROM status read mode

  The FCU enters this mode when the FCU accepts a command that is neither a normal mode transition command nor a lock bit read mode transition command in ROM P/E mode. The ROM status read mode includes the state in which the FRDY bit in FSTATR0 is 0 and the command-locked state after an error has occurred. Table 26.13 shows the commands that can be accepted in this mode. High-speed read operation is not available for the ROM. If an address in the range from H'80BC0000 to H'80BFFFFF is read through the P-bus while the FENTRY6 bit is set to 1, an address in the range from H'80B80000 to H'80BBFFFF is read through the P-bus while the FENTRY5 bit is set to 1, an address in the range from H'80B40000 to H'80B7FFFF is read through the P-bus while the FENTRY4 bit is set to 1, an address in the range from H'80B00000 to H'80B3FFFF is read through the P-bus while the FENTRY3 bit is set to 1, an address in the range from H'80A00000 to H'80AFFFFF is read through the P-bus while the FENTRY2 bit is set to 1, an address in the range from H'80900000 to H'809FFFFF is read through the P-bus while the FENTRY1 bit is set to 1, or an address in the range from H'80800000 to H'808FFFFF is read through the P-bus while the FENTRY0 bit is set to 1, the FSTATR0 value is read.

- ROM lock bit read mode

  The FCU enters this mode when the FCU accepts a lock bit read mode transition command in ROM P/E mode. Table 26.13 shows the commands that can be accepted in this mode. High-speed read operation is not available for the ROM. The FENTRYR value is the same as that in ROM P/E normal mode. If an address in the range from H'80BC0000 to H'80BFFFFF is read through the P-bus while the FENTRY6 bit is set to 1, an address in the range from H'80B80000 to H'80BBFFFF is read through the P-bus while the FENTRY5 bit is set to 1, an address in the range from H'80B40000 to H'80B7FFFF is read through the P-bus while the FENTRY4 bit is set to 1, an address in the range from H'80B00000 to H'80B3FFFF is read through the P-bus while the FENTRY3 bit is set to 1, an address in the range from H'80A00000 to H'80AFFFFF is read through the P-bus while the FENTRY2 bit is set to 1, an address in the range from H'80900000 to H'809FFFFF is read through the P-bus while the FENTRY1 bit is set to 1, or an address in the range from H'80800000 to H'808FFFFF is read through the P-bus while the FENTRY0 bit is set to 1, the lock bit value of the target erasure block is returned through all bits in the read data.

Table 26.13 shows the acceptable commands in each FCU mode/state. When a command that cannot be accepted is issued, the FCU enters the command-locked state (see section 26.8.3, Error Protection).

To make sure that the FCU accepts a command, enter the mode in which the FCU can accept the target command, check the FRDY, ILGLERR, ERSERR, and PRGERR bit values in FSTATR0, and the FCUERR, FRDTCT, and FRCRCT bit values in FSTATR1, and then issue the target FCU command. The CMDLK bit in FASTAT holds a value obtained by logical ORing the ILGLERR, ERSERR, and PRGERR bit values in FSTATR0 and the FCUERR, FRDTCT, and FRCRCT bit values in the FSTATR1. Therefore the FCU's error occurrence state can be checked by reading the CMDLK bit. In table 26.13, the CMDLK bit is used as the bit to indicate the error occurrence state. The FRDY bit of FSTATR0 is 0 during the programming/erasure, programming/erasure suspension, and lock bit read 2 processes. While the FRDY bit is 0, the P/E suspend command can be accepted only when the SUSRDY bit in FSTATR0 is 1.

Table 26.13 includes 0 and 1 in single cells of the ERSSPD, PRGSPD, and FRDY bit rows for the sake of simplification. The ERSSPD bits 1 and 0 indicate the erasure suspension and programming suspension processes, respectively. The PRGSPD bits 1 and 0 indicate the programming suspension and erasure suspension processes, respectively. The FRDY bit value can be either 1 or 0, which is a value held by the bit prior to a transition to the command lock state.

Table 26.13 FCU Modes/States and Acceptable Commands (ROM P/E Mode)

	P/E Normal Mode				Status Read Mode										Lock Bit Read Mode			
Item	Programming-Suspended	Erasure-Suspended	Protected-Locked State	Other State*1	Programming/Erasure Processing	Programming Processing during Erasure-Suspended	Programming/Erasure Suspension Processing	Lock Bit Read 2 Processing	Programming-Suspended	Erasure-Suspended	Command-Locked (FRDY = 0)	Command-Locked (FRDY = 1)	Protected-Locked State	Other State*1	Programming-Suspended	Erasure-Suspended	Protected-Locked State	Other State*1
FRDY bit in FSTATR0	1	1	1	1	0	0	0	0	1	1	0	1	1	1	1	1	1	1
SUSRDY bit in FSTATR0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
ERSSPD bit in FSTATR0	0	1	0	0	0	1	0/1	0/1	0	1	0/1	0/1	0	0	0	1	0	0
PRGSPD bit in FSTATR0	1	0	0	0	0	0	0/1	0/1	1	0	0/1	0/1	0	0	1	0	0	0
CMDLK bit in FASTAT	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
IFSST bits in IFSSR	Other than 101/111	Other than 101/111	101/111	Other than 101/111	Other than 101/111	Other than 101/111	Other than 101/111	Other than 101/111	Other than 101/111	Other than 101/111	*2	*2	101/111	Other than 101/111	Other than 101/111	Other than 101/111	101/111	Other than 101/111
Normal mode transition	A	A	A	A	×	×	×	×	A	A	×	×	A	A	A	A	A	A
Status read mode transition	A	A	A	A	×	×	×	×	A	A	×	×	A	A	A	A	A	A
Lock bit read mode transition (lock bit read 1)	A	A	×	A	×	×	×	×	A	A	×	×	×	A	A	A	×	A
Program	×	*	×	A	×	×	×	×	×	*	×	×	×	A	×	*	×	A
Block erase	×	×	×	A	×	×	×	×	×	×	×	×	×	A	×	×	×	A
P/E suspend	×	×	×	×	A	×	×	×	×	×	×	×	×	×	×	×	×	×
P/E resume	A	A	×	×	×	×	×	×	A	A	×	×	×	×	A	A	×	×
Status register clear	A	A	×	A	×	×	×	×	A	A	×	A	×	A	A	A	×	A
Lock bit read 2	A	A	×	A	×	×	×	×	A	A	×	×	×	A	A	A	×	A
Lock bit program	×	*	×	A	×	×	×	×	×	*	×	×	×	A	×	*	×	A

[Legend]
A: Acceptable
*: Only programming is acceptable for the areas other than the erasure-suspended block
×: Not acceptable

Notes: *1 Including unprotected state / protected-unlocked state
*2 Not depending on the state of the IFSST bits

26.6.3 FCU Command Usage

This section shows examples of user processing procedures for firmware transfer to the FCU RAM and the issuing of FCU commands. In some procedures given in this section, the FCU state is not checked before an FCU command is issued but the command result is checked before the processing is completed. To make sure that the FCU accepts a command, check the FCU state before starting processing (see section 26.6.2, Conditions for FCU Command Acceptance).

In a flow used in this section, the current state of FCU command handling and error occurrence is checked via the FRDY, ILGLERR, ERSERR, PRGERR, SUSRDY, ERSSPD, and PRGSPD bits in FSTATR0 and the FCUERR, FRDTCT, and FRCRCT bits in FSTATR1. Since both FSTATR0 and FSTATR1 can be read in word access at a time, the FCU state can be checked by making register access only once. If the FCU state is checked via the FRDY bit of FSTATR0 and the CMDLK bit of FASTAT, register access must be made twice. However, the state of error occurrence can be checked via the CMDLK bit only.

The FRDY bit retains 0, if the FRDTCT and FRCRCT bits are set to 1 to put the FCU into a command-locked state in the middle of its command handling while the FCUERR bit is 1 or the FRDCLE and FRCCLE bits are 1. Since the FCU in a command-locked state halts its processes, the FRDY bit is never set to 1 from 0. If the FRDY retains 0 for a longer period than programming/erasing time or suspend delay time (see section 34, Electrical Characteristics), abnormal operation such as the FCU process halt may have occurred. In such case, initialize the FCU by an FCU reset. If the FRDY is set to 1 upon completion of the FCU command handling, while the FRDCLE and FRCCLE bits are 1, the FCUERR bit and the FRDTCT and FRCRCT bits are also 0. Therefore, the state of error occurrence can be checked via the ILGLERR, ERSERR, and PRGERR bits.

(1) Transferring Firmware to the FCU RAM

To use FCU commands, the FCU firmware must be stored in the FCU RAM. When this LSI is started, the FCU firmware is not stored in the FCU RAM; copy the firmware stored in the FCU firmware area to the FCU RAM. If the FCUERR, FRDTCT, or FRCRCT bit in FSTATR1 is 1, the firmware stored in the FCU RAM may have been damaged; reset the FCU and copy the FCU firmware again in this case.

Figure 26.14 shows the procedure for firmware transfer to the FCU RAM. Before writing data to the FCU RAM, clear FENTRYR to H'0000 to stop the FCU. For details on the DMAC settings, refer to section 11, Direct Memory Access Controller (DMAC).

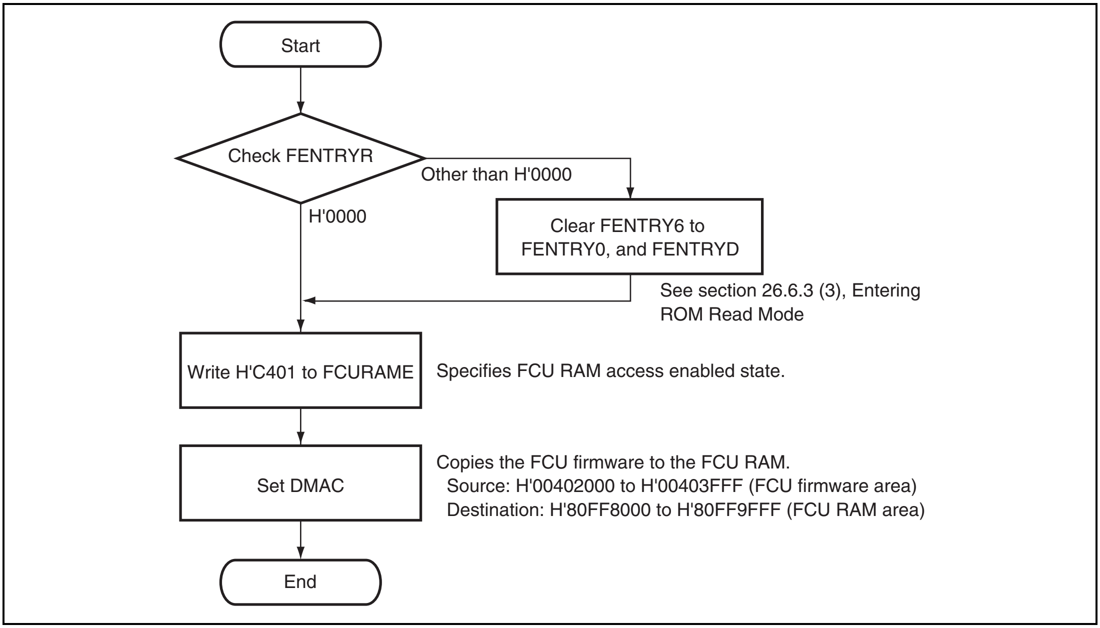
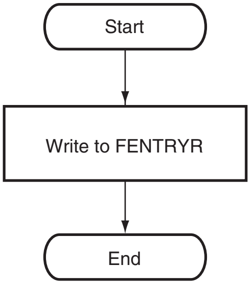

Start
Check FENTRYR
Other than H'0000
H'0000
Clear FENTRY6 to FENTRY0, and FENTRYD
See section 26.6.3 (3), Entering ROM Read Mode
Write H'C401 to FCURAME
Specifies FCU RAM access enabled state.
Set DMAC
Copies the FCU firmware to the FCU RAM.
Source: H'00402000 to H'00403FFF (FCU firmware area)
Destination: H'80FF8000 to H'80FF9FFF (FCU RAM area)
End

Figure 26.14 Procedure for Firmware Transfer to FCU RAM

(2) Entering ROM P/E Mode

To execute ROM-related FCU commands, set the FENTRY6 to FENTRY0 bits in FENTRYR appropriately to make the FCU enter ROM P/E mode (see section 26.6.2, Conditions for FCU Command Acceptance). To execute FCU commands for the ROM, set the corresponding FENTRY6 to FENTRY0 bits to 1. For the conditions for writing to the FENTRY6 to FENTRY0 bits, refer to section 26.3.10, Flash P/E Mode Entry Register (FENTRYR).

After a transition from ROM read mode to ROM P/E mode, the FCU is in ROM P/E normal mode.

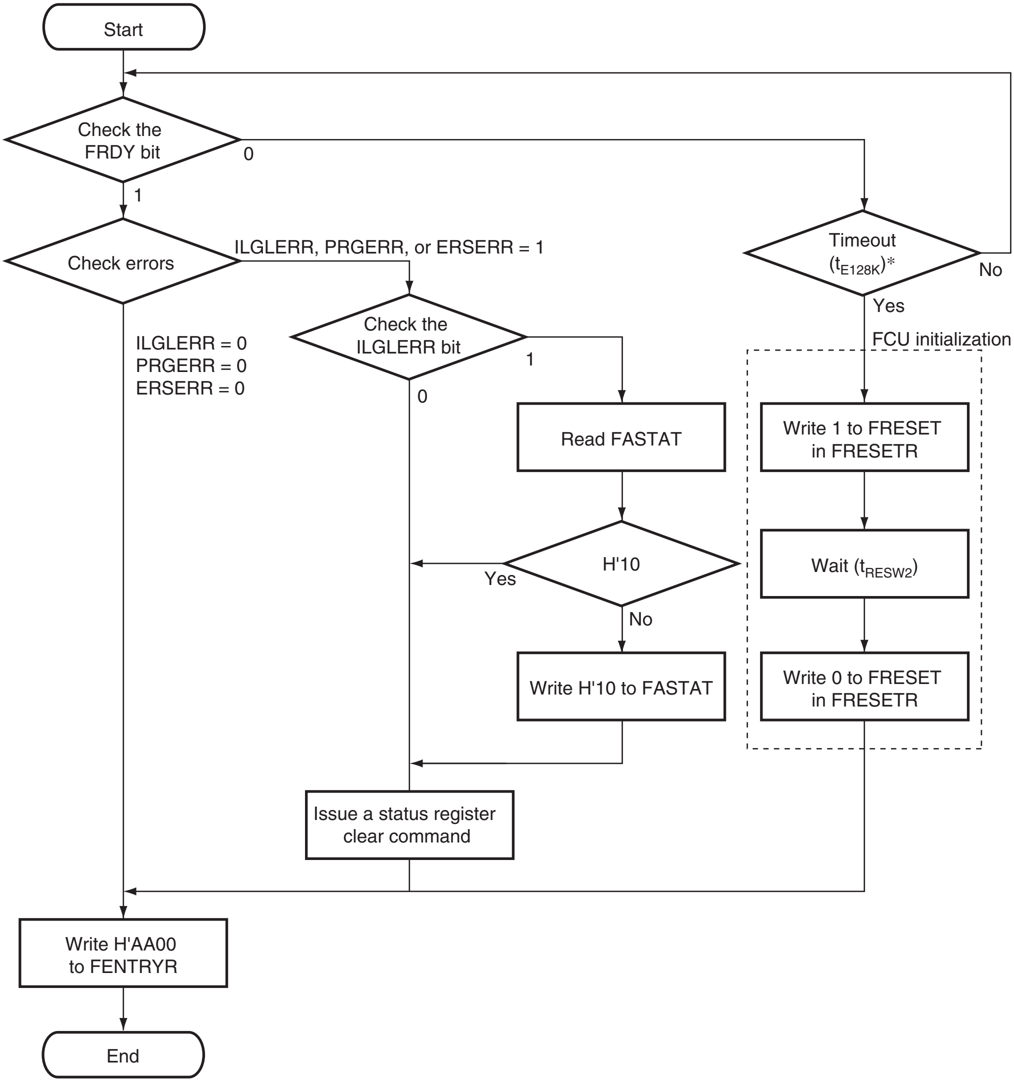


Specifies ROM P/E mode.
To set FENTRY6 to 1: Write H'AA40.
To set FENTRY5 to 1: Write H'AA20.
To set FENTRY4 to 1: Write H'AA10.
To set FENTRY3 to 1: Write H'AA08.
To set FENTRY2 to 1: Write H'AA04.
To set FENTRY1 to 1: Write H'AA02.
To set FENTRY0 to 1: Write H'AA01.

Figure 26.15 Procedure for Transition to ROM P/E Mode

(3) Entering ROM Read Mode

To enable high-speed ROM read access through the ROM cache, clear the FENTRY6 to FENTRY0 bits in FENTRYR to make the FCU enter ROM read mode (see section 26.6.2, Conditions for FCU Command Acceptance). A transition from ROM P/E mode to ROM read mode must be made while no FCU error has been detected since FCU command processing is completed.



Notes: t_{E128K} : Time required to erase a 128-kbyte block (see section 34, Electrical Characteristics).
t_{RESW2}: Reset pulse width during programming and erasure (see section 34, Electrical Characteristics).

Figure 26.16 Procedure for Transition to ROM Read Mode

(4) Using ROM P/E Normal Mode Transition Command

The FCU can be moved to ROM P/E normal mode in two ways: one is to set FENTRYR appropriately in ROM read mode (see section 26.6.3 (1), Transferring Firmware to the FCU RAM) and the other is to issue a normal mode transition command in ROM P/E mode (figure 26.17). The status read mode transition command and the lock bit read mode transition command can be used in the same way as the normal mode transition command.

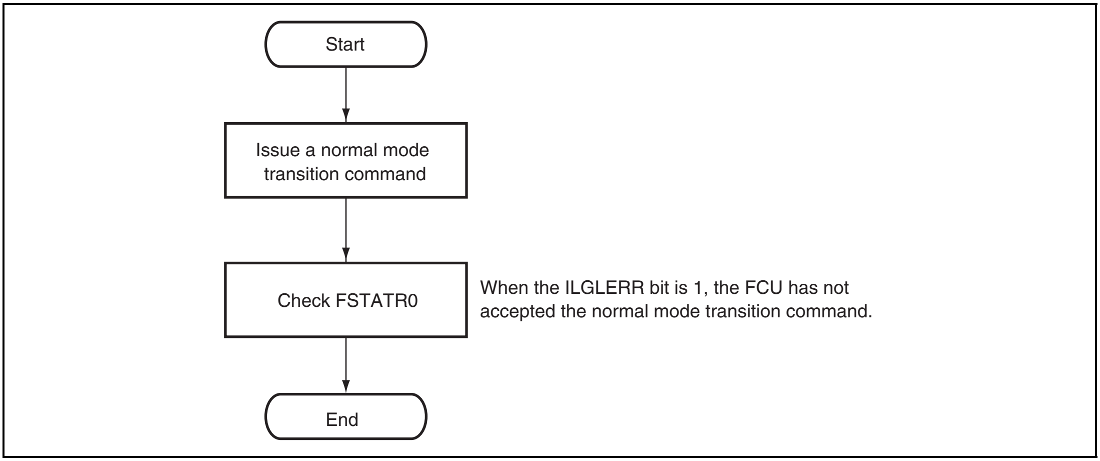

Start
Issue a normal mode transition command
Check FSTATR0
When the ILGLERR bit is 1, the FCU has not accepted the normal mode transition command.
End

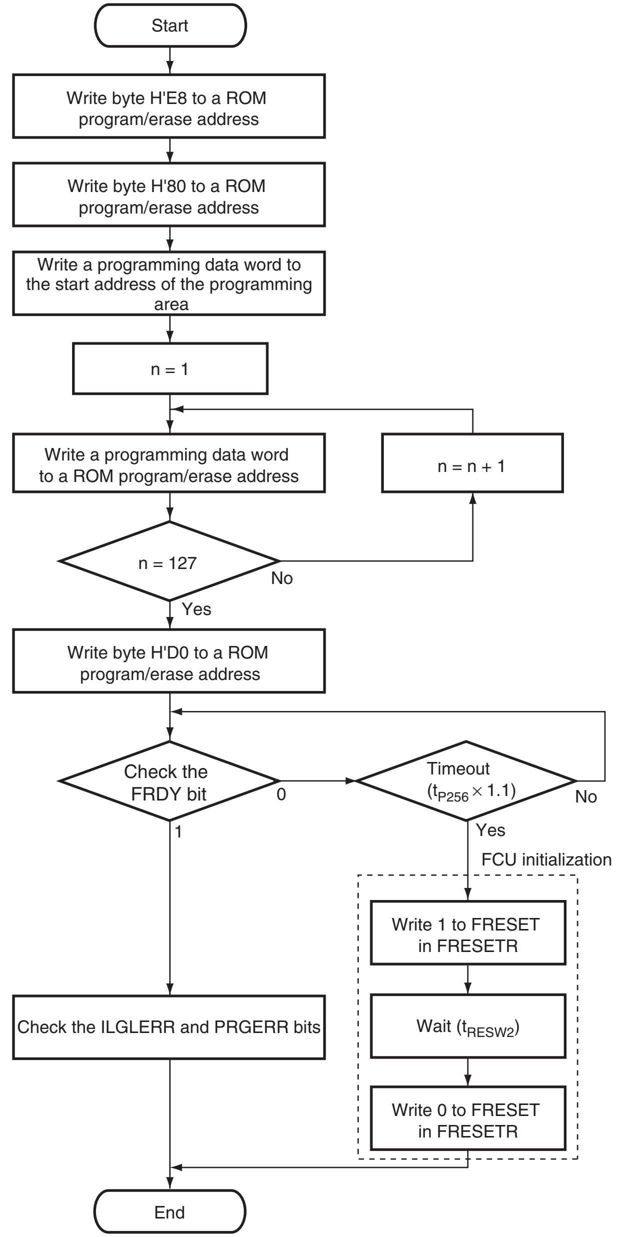
Figure 26.17 Procedure to Use ROM P/E Normal Mode Transition Command

(5) Programming

To program the ROM, use the program command. Write byte H'E8 to a ROM program/erase address in the first cycle of the program command and byte H'80 in the second cycle. Access the P bus in words from the third to 130th cycles of the command. In the third cycle, write the programming data to the start address of the target programming area. Here, the start address must be a 256-byte boundary address. After writing words to ROM program/erase addresses 127 times, write byte H'D0 to a ROM program/erase address in the 131st cycle; the FCU then starts ROM programming. Read the FRDY bit in FSTATR0 to confirm that ROM programming is completed.

The addresses that can be specified in the first to 131st cycles depend on the setting of FENTRY6 to FENTRY0 bits in FENTRYR. An address in the range from H'80BC0000 to H'80BFFFFF can be specified when the FENTRY6 bit is set to 1, an address in the range from H'80B80000 to H'80BBFFFF can be specified when the FENTRY5 bit is set to 1, an address in the range from H'80B40000 to H'80B7FFFF can be specified when the FENTRY4 bit is set to 1, an address in the range from H'80B00000 to H'80B3FFFF can be specified when the FENTRY3 bit is set to 1, an address in the range from H'80A00000 to H'80AFFFFF can be specified when the FENTRY2 bit is set to 1, an address in the range from H'80900000 to H'809FFFFF can be specified when the FENTRY1 bit is set to 1, or an address in the range from H'80800000 to H'808FFFFF can be specified when the FENTRY0 bit is set to 1. If a command is issued while an illegal combination of FENTRY6 to FENTRY0 bit values and addresses is specified, the FCU detects an error and enters command-locked state (see section 26.8.3, Error Protection).

If the area accessed in the third to 130th cycles includes addresses that do not need to be programmed, write H'FFFF as the programming data for those addresses. To ignore the protection provided by the lock bit during programming, set the FPROTCN bit in FPROTR to 1 before starting programming.


Start
Write byte H'E8 to a ROM program/erase address
Write byte H'80 to a ROM program/erase address
Write a programming data word to the start address of the programming area
n = 1
Write a programming data word to a ROM program/erase address
n = n + 1
n = 127
No
Yes
Write byte H'D0 to a ROM program/erase address
Check the FRDY bit
0
1
Timeout (t_{P256} × 1.1)
No
Yes
FCU initialization
Write 1 to FRESET in FRESETR
Wait (t_{RESW2})
Write 0 to FRESET in FRESETR
Check the ILGLERR and PRGERR bits
End

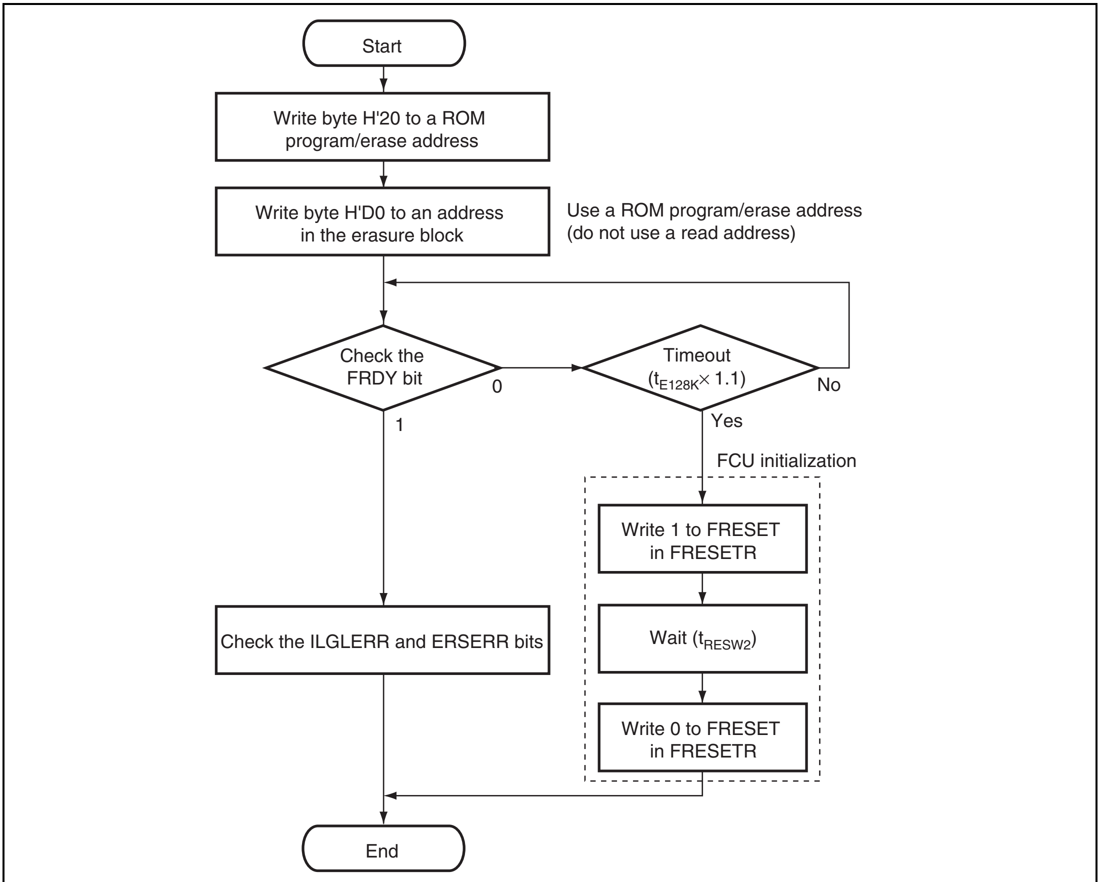
Notes: t_{P256}: Time required to write 256-byte data (see section 34, Electrical Characteristics).
t_{RESW2}: Reset pulse width during programming and erasure (see section 34, Electrical Characteristics).

Figure 26.18 Procedure for ROM Programming

(6) Erasure

To erase the ROM, use the block erase command. Write byte H'20 to a ROM program/erase address in the first cycle of the block erase command. Write byte H'D0 to an address in the target erasure block in the second cycle; the FCU then starts ROM erasure. Read the FRDY bit in FSTATR0 to confirm that ROM erasure is completed.

To ignore the protection provided by the lock bit during erasure, set the FPROTCN bit in FPROTR to 1 before starting erasure.


Start
Write byte H'20 to a ROM program/erase address
Write byte H'D0 to an address in the erasure block
Use a ROM program/erase address (do not use a read address)
Check the FRDY bit
0
1
Timeout (t_{E128K}× 1.1)
No
Yes
FCU initialization
Write 1 to FRESET in FRESETR
Wait (t_{RESW2})
Write 0 to FRESET in FRESETR
Check the ILGLERR and ERSERR bits
End

Notes: t_{E128K}: Time required to erase a 128-kbyte block (see section 34, Electrical Characteristics).
t_{RESW2}: Reset pulse width during programming and erasure (see section 34, Electrical Characteristics).

Figure 26.19 Procedure for ROM Erasure

(7) Suspending Programming or Erasure

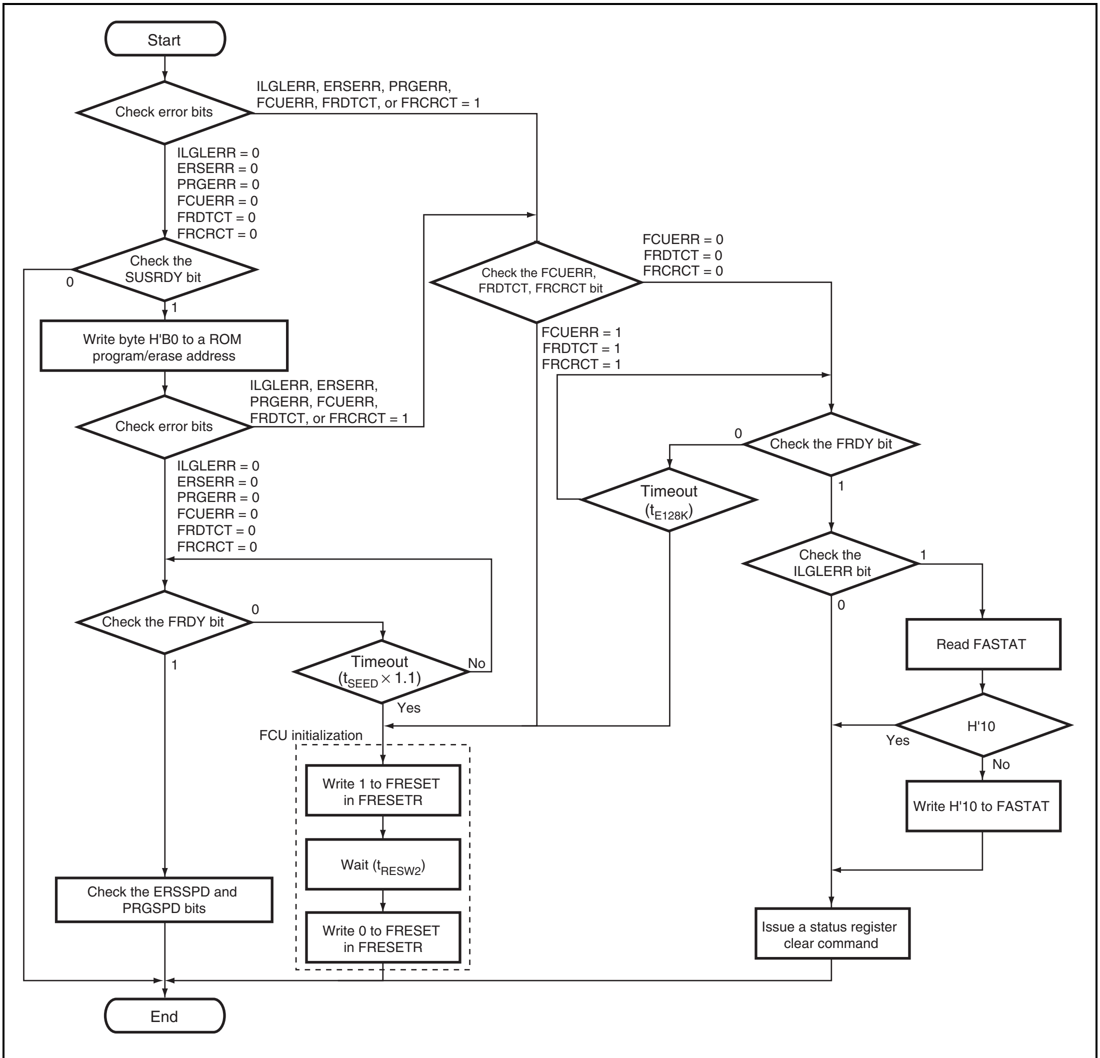
To suspend programming or erasure of the ROM, use the P/E suspend command. Before issuing a P/E suspend command, check that the ILGLERR, ERSERR, and PRGERR bits in FSTATR0 and the FCUERR, FRDTCT, and FRCRCT bits in FSTATR1 are 0; that is, to ensure that programming or erasure processing is being performed correctly. Also, check that the SUSRDY bit in FSTATR1 is 1 to ensure that a suspend command is acceptable.

After issuing a P/E suspend command, read both FSTATR0 and FSTATR1 to ensure no error has occurred. If an error has occurred, at least one of the ILGLERR, PRGERR, ERSERR, FCUERR, FRDTCT, and FRCRCT bits is set to 1. If programming/erasure is complete within the period from when the SUSRDY bit is ensured to be 1 until a P/E suspend command is accepted, the ILGLERR bit is set to 1 as the issued command is detected as illegal. If a P/E suspend command is accepted when programming/erasure is complete, no error occurs, hence no transition to a suspended state (the FRDY bit is 1 and both the ERSSPD and PRGSPD bits are 0).

Once a P/E suspend command is accepted and programming/erasure is normally suspended, the FCU enters a suspended state and that the FRDY bit is 1 and the ERSSPD or PRGSPD bit is 1. After issuing a P/E suspend and ensuring that the FCU has entered a suspend state, determine which operation to perform in the succeeding process. If a P/E resume command is issued in the succeeding process while the FCU has not entered a suspended state, an illegal command error occurs and the FCU enters a command-locked state (see section 26.8.3, Error Protection).

Once the FCU has entered the erasure-suspended state, blocks not for erasing can be written to. In both programming-suspended and erasure-suspended states, the FCU can be moved to ROM read mode by clearing FENTRYR.

For the operation when the FCU accepts a P/E suspend command, see section 26.6.4, Suspending Operation.

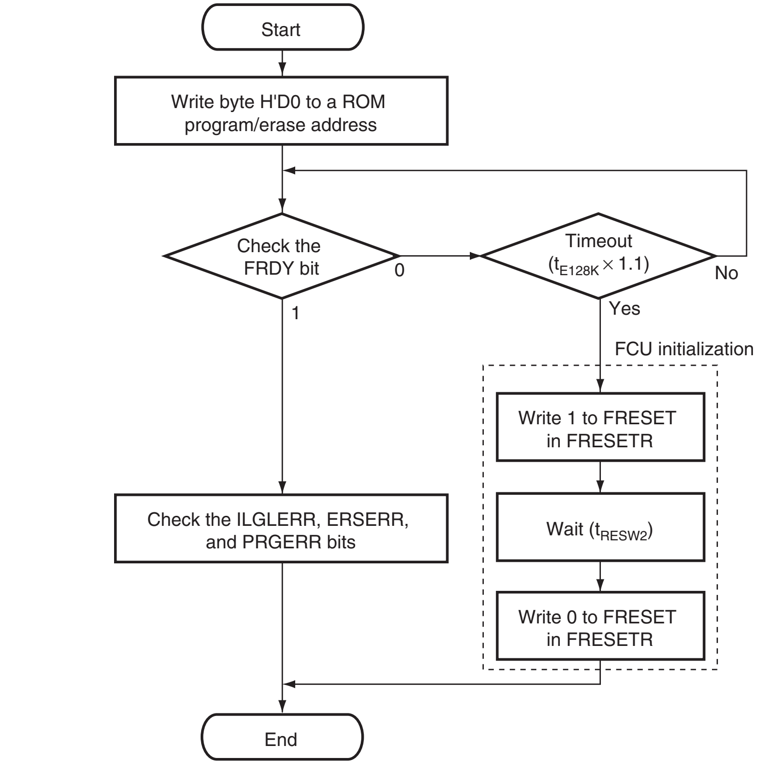


Notes: t_{SEED} : Suspension delay time.
t_{E128K} : Time required to erase a 128-kbyte block (see section 34, Electrical Characteristics).

Figure 26.20 Procedure for Programming/Erasure Suspension

(8) Resuming Programming or Erasure

To resume programming or erasure that has been suspended, use the P/E resume command. If the FENTRYR setting has been modified during suspension, issue a P/E resume command only after resetting FENTRYR to the previous value that was held before the P/E suspension command was issued.


Start
Write byte H'D0 to a ROM program/erase address
Check the FRDY bit
0
1
Timeout (t_{E128K} × 1.1)
No
Yes
FCU initialization
Write 1 to FRESET in FRESETR
Wait (t_{RESW2})
Write 0 to FRESET in FRESETR
Check the ILGLERR, ERSERR, and PRGERR bits
End

Notes: t_{E128K} : Time required to erase a 128-kbyte block (see section 34, Electrical Characteristics).
t_{RESW2}: Reset pulse width during programming and erasure (see section 34, Electrical Characteristics).

Figure 26.21 Procedure for Resuming Programming or Erasure

(9) Clearing Status Register 0 (FSTATR0)

To clear the ILGLERR, PRGERR, and ERSERR bits in FSTATR0, use the status register clear command. When any one of the ILGLERR, PRGER, and ERSERR bits is 1, the FCU is in command-locked state, in which the FCU only accepts the status register clear command and does not accept other commands. When the ILGLERR bit is 1, check also the value of the ROMAE, EEPAE, EEPIFE, EEPRPE, and EEPWPE bits in FASTAT. If a status register clear command is issued without clearing these bits, the ILGLERR bit is not cleared.

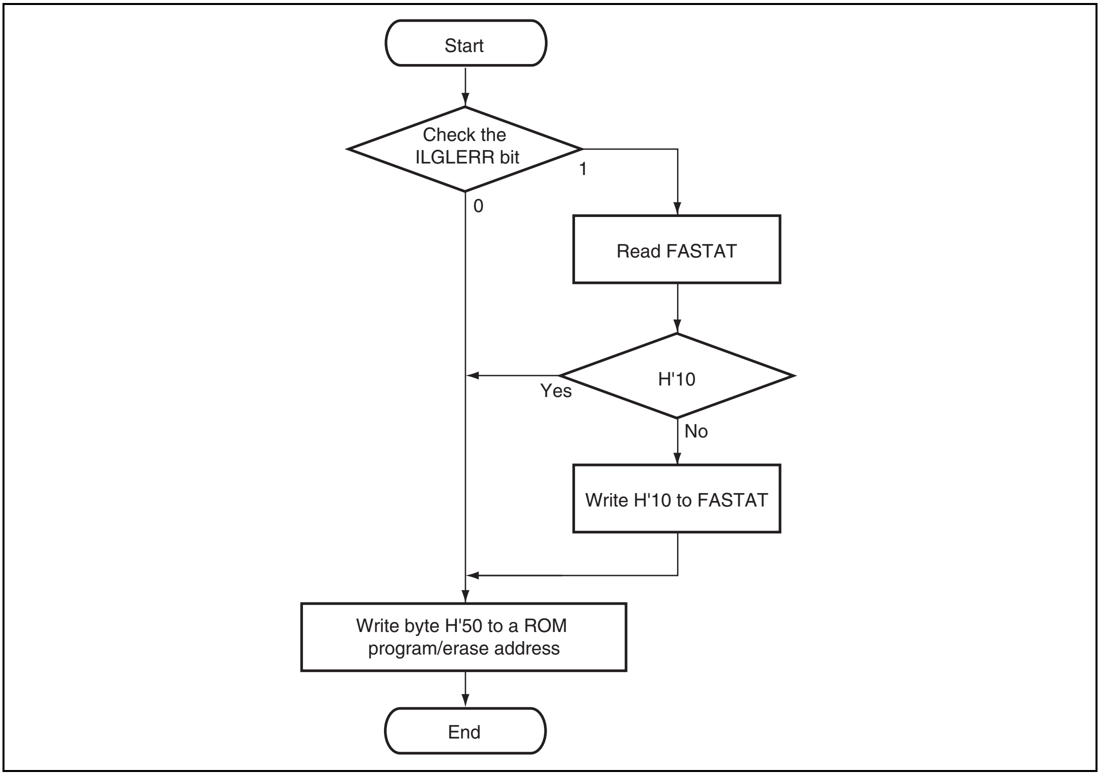


Figure 26.22 Procedure for Clearing Status Register 0

(10) Checking Status Register 0 (FSTATR0)

The FSTATR0 value can be checked in two ways: one is to directly read FSTATR0 and the other is to read a ROM program/erase address in ROM status read mode. After an FCU command is issued that is neither a normal mode transition command nor a lock bit read mode transition command, the FCU is in ROM status read mode. In the example shown in figure 26.23, a status read mode transition command is issued to enter ROM status read mode, and then a ROM program/erase address is read to check the FSTATR0 value.

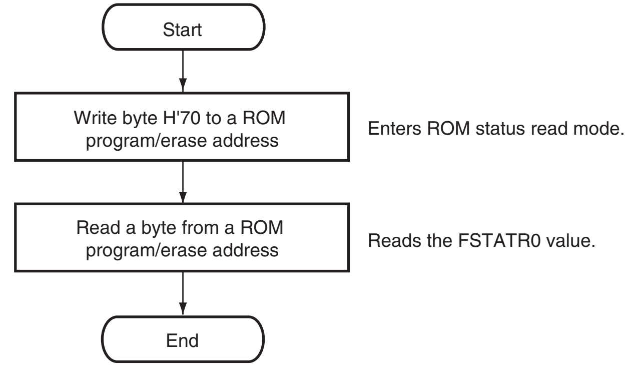

Start
Write byte H'70 to a ROM program/erase address
Enters ROM status read mode.
Read a byte from a ROM program/erase address
Reads the FSTATR0 value.
End

Figure 26.23 Procedure for Checking Status Register 0

(11) Reading Lock Bit

Each erasure block in the user MAT has a lock bit. While the FPROTCN bit in FPROTR is 0, the erasure block whose lock bit is set to 0 cannot be programmed or erased.

The lock bit status can be checked in either memory area read mode or register read mode. In memory area read mode (the FRDMD bit in FMODR is 0), read a ROM program/erase address in ROM lock bit read mode, and the lock bit value in the specified erasure block is copied to all bits in the data read through the P bus. In register read mode (the FRDMD bit in FMODR is 1), issue a lock bit read 2 command, and the lock bit value in the specified erasure block is copied to the FLOCKST bit in FSTATR1.

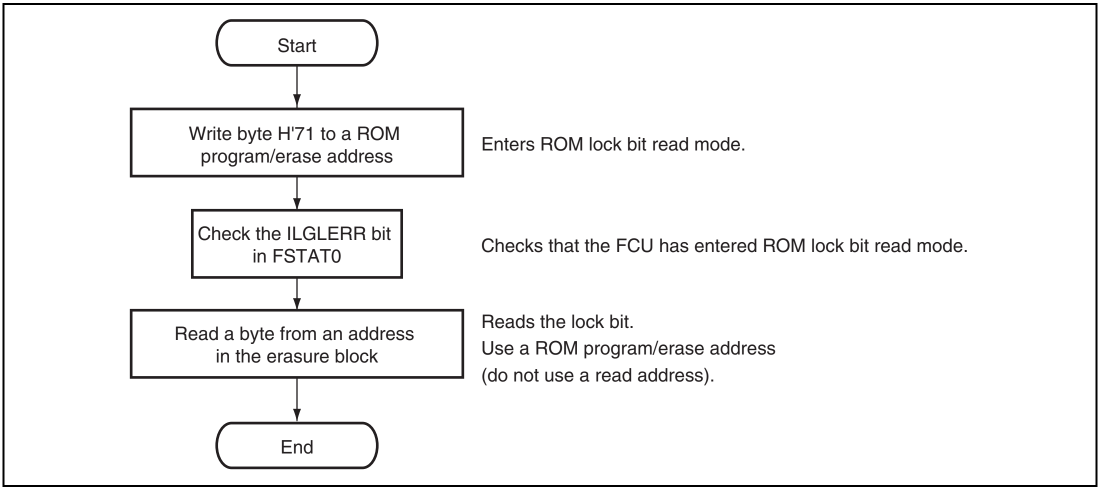
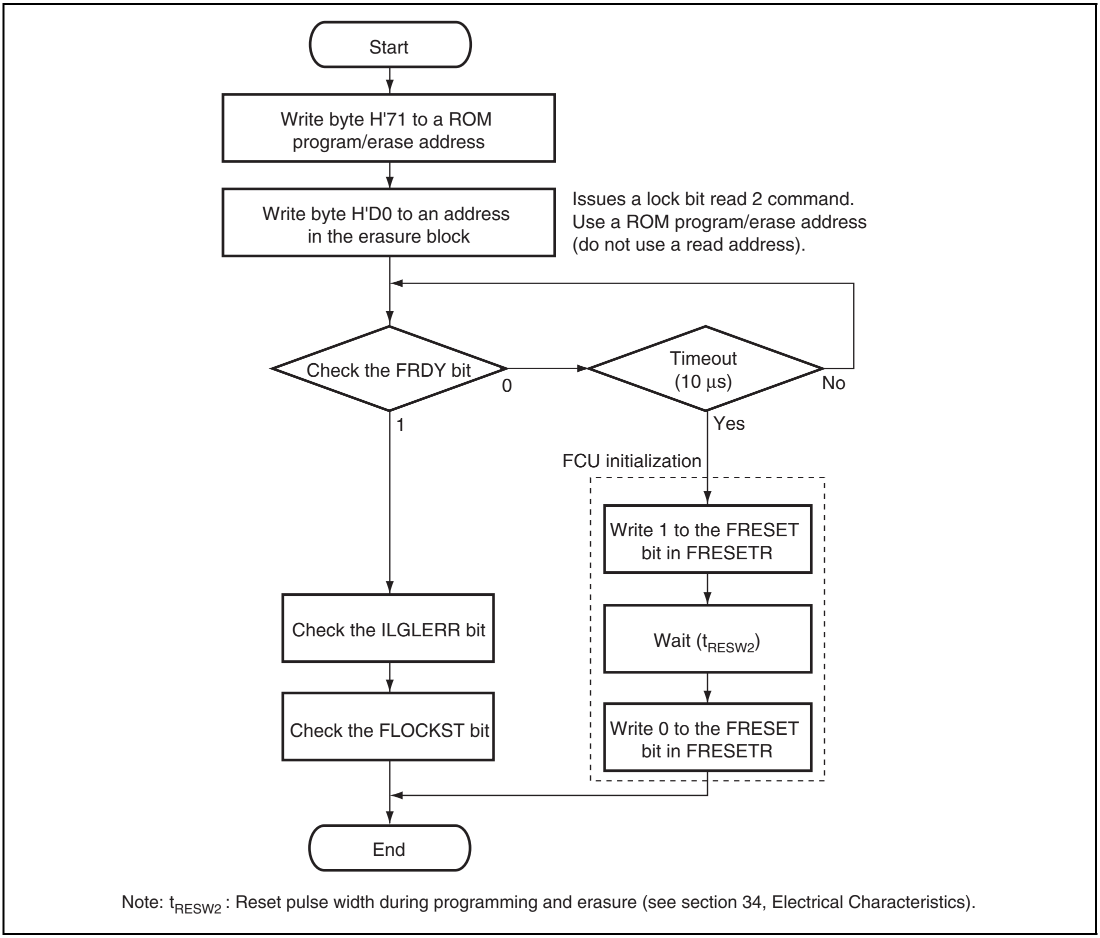

Start
Write byte H'71 to a ROM program/erase address
Enters ROM lock bit read mode.
Check the ILGLERR bit in FSTAT0
Checks that the FCU has entered ROM lock bit read mode.
Read a byte from an address in the erasure block
Reads the lock bit.
Use a ROM program/erase address
(do not use a read address).
End

Figure 26.24 Procedure for Reading Lock Bit in Memory Area Read Mode

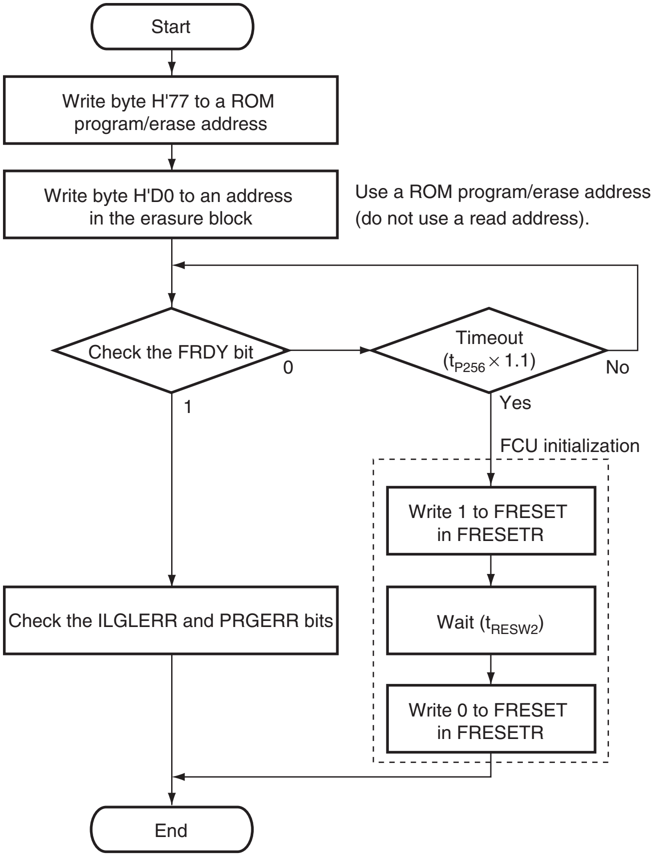

Start
Write byte H'71 to a ROM program/erase address
Write byte H'D0 to an address in the erasure block
Issues a lock bit read 2 command.
Use a ROM program/erase address
(do not use a read address).
Check the FRDY bit
0
1
Timeout
(10 µs)
No
Yes
FCU initialization
Write 1 to the FRESET bit in FRESETR
Wait (t_{RESW2})
Write 0 to the FRESET bit in FRESETR
Check the ILGLERR bit
Check the FLOCKST bit
End

Note: t_{RESW2} : Reset pulse width during programming and erasure (see section 34, Electrical Characteristics).

Figure 26.25 Procedure for Reading Lock Bit in Register Read Mode

(12) Writing to Lock Bit

Each erasure block in the user MAT has a lock bit. To write to a lock bit, use the lock bit program command. Write byte H'77 to a ROM program/erase address in the first cycle of the lock bit program command. Write byte H'D0 to an address in the target erasure block whose lock bit is to be written to in the second cycle; the FCU then starts writing to the lock bit. Read the FRDY bit in FSTATR0 to confirm that writing is completed.


Start
Write byte H'77 to a ROM program/erase address
Write byte H'D0 to an address in the erasure block
Use a ROM program/erase address (do not use a read address).
Check the FRDY bit
0
1
Timeout (t_{P256} × 1.1)
No
Yes
FCU initialization
Write 1 to FRESET in FRESETR
Wait (t_{RESW2})
Write 0 to FRESET in FRESETR
Check the ILGLERR and PRGERR bits
End

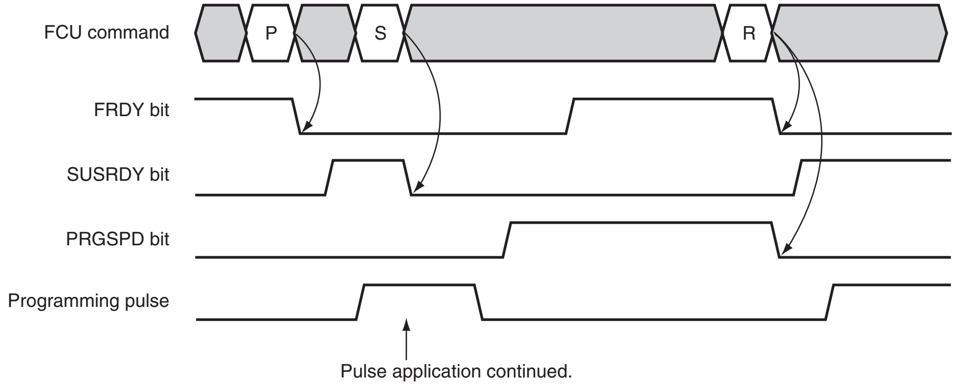
Notes: t_{P256}: Time required to write 256-byte data (see section 34, Electrical Characteristics).
t_{RESW2}: Reset pulse width during programming and erasure (see section 34, Electrical Characteristics).

Figure 26.26 Procedure for Writing to the Lock Bit

To erase a lock bit, use the block erase command. While the FPROTCN bit in FPROTR is 0, the erasure block whose lock bit is set to 0 cannot be erased. Set the FPROTCN bit to 1, and then issue a block erase command to erase a lock bit. The block erase command erases all data in the specified erasure block; it is not possible to erase only the lock bit.

26.6.4 Suspending Operation

When a P/E suspend command is issued while ROM is being programmed or erased, the FCU suspends the programming or erasure processing. Figure 26.27 gives an overview of operation for suspending programming. Upon accepting a programming command, the FCU clears the FRDY bit in FSTATR0 to 0 and starts programming. Once the FCU enters a state where it is ready to accept a command after the start of programming, the SUSRDY bit is set to 1. If a P/E suspend command is issued, the FCU accepts the command and clears the SUSRDY bit. If the FCU accepts the command while reapplying a write pulse, the FCU continues applying the pulse. After a specified pulse application time has elapsed, the FCU completes applying the pulse, suspends programming, and sets the PRGSPD bit to 1. Once the process completes, the FCU sets the FRDY bit to 1 and enters a programming suspended state. If the FCU accepts a P/E resume command in this state, the FCU clears the FRDY and PRGSPD bits to 0 and restarts programming.

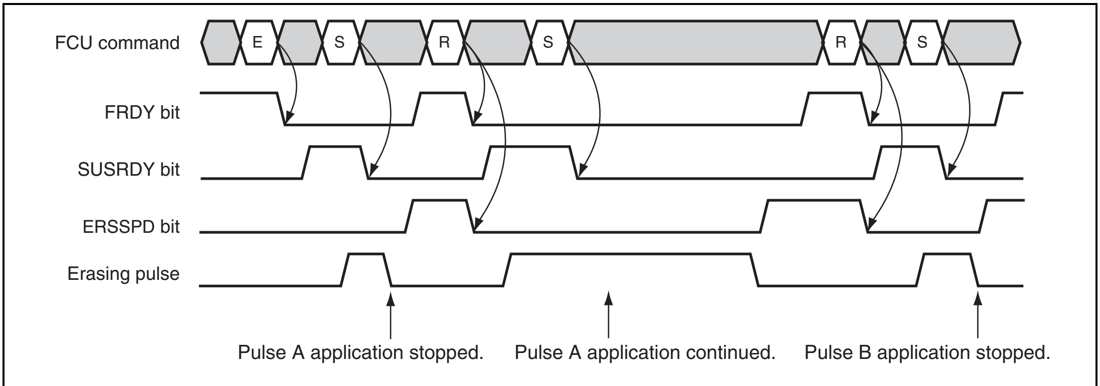


[Legend]
P: Programming command (interleaved program, lock bit program, or P/E resume command)
S: P/E suspend command
R: P/E resume command

Figure 26.27 Suspending Programming Processing

Figure 26.28 shows the operation for suspending erasure processing in suspension-priority mode (the ESUSPMD bit in FCPSR is 0). Upon accepting an erasing command, the FCU clears the FRDY bit to 0 and starts erasing. Once the FCU enters a state where it is ready to accept a command after the start of erasing, the SUSRDY bit is set to 1. If a P/E suspend command is issued, the FCU accepts the command and clears the SUSRDY bit. If the FCU accepts the command during its erasing operation, the FCU starts a suspending process even while applying a pulse and sets the ERSSPD bit to 1. Once the suspending process completes, the FCU sets the FRDY bit to 1 and enters an erasing suspended state. If the FCU accepts a P/E resume command in this state, the FCU clears the FRDY and PRGSPD bits to 0 and restarts erasing. The operations of the FRDY, SUSRDY, and ERSSPD bits are independent of the erasure-suspended mode.

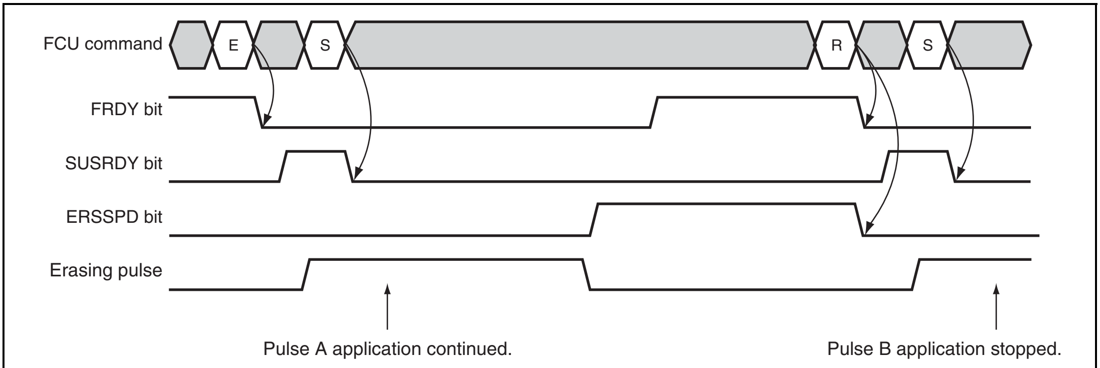
The setting for the erasure-suspended mode affect the control methods for erasure pulse. In suspend-priority mode, if the FCU accepts a P/E suspend command while applying erasure pulse A, which has not been suspended previously, the FCU suspends the pulse application and enters an erasure-suspended state. After the FCU resumes erasing by accepting a P/E resume command, if the FCU accepts a P/E suspend command while applying erasing pulse A, the FCU continues applying the pulse. After a specified pulse application time has elapsed, the FCU completes applying the pulse and enters an erasure-suspended state. Next, after the FCU accepts a P/E resume command and starts applying a new pulse B, if the FCU accepts a P/E suspend command, the FCU suspends the pulse application. In suspense-priority mode, the suspense process is given priority by suspending once every pulse application.


FCU command
E
S
R
S
R
S
FRDY bit
SUSRDY bit
ERSSPD bit
Erasing pulse
Pulse A application stopped.
Pulse A application continued.
Pulse B application stopped.

[Legend]
P: Erasing command (block erase or P/E resume command)
S: P/E suspend command
R: P/E resume command

Figure 26.28 Suspending Erasure Processing (Suspension-Priority Mode)

The operation for suspending erasure processing in erasure-priority mode (the ESUSPMD bit in FCPSR is 1) is equivalent to that for suspending programming processing. In erasure-priority mode, if the FCU accepts a P/E suspend command while applying an erasing pulse, the FCU always continues applying the pulse. As processing to reapply an erasing pulse never takes place in this mode, the total time required for erasure processing is shorter than in suspension-priority mode.



[Legend]
P: Erasing command (block erase or P/E resume command)
S: P/E suspend command
R: P/E resume command

Figure 26.29 Suspending Erasure Processing (Erasure-Priority Mode)

26.7 User Boot Mode

To program or erase the user MAT in user boot mode, issue FCU commands to the FCU. A user-defined boot mode can be implemented by writing to the user boot MAT a ROM programming/erasing routine that uses a desired communications interface; when this LSI is started in user boot mode after that, the user-defined boot mode is initiated. Programming/erasure of the user boot MAT is only enabled in boot mode.

26.7.1 User Boot Mode Initiation

When this LSI is started in user boot mode, execution starts in the embedded program stored MAT, necessary processing such as FCU firmware transfer to the FCU RAM is performed, and then execution jumps to the location indicated by the reset vector of the user boot MAT. Figure 26.30 gives an overview of the boot sequence.
Refer to section 27, Intelligent Flash Security (IFS), for details of key-code approval.

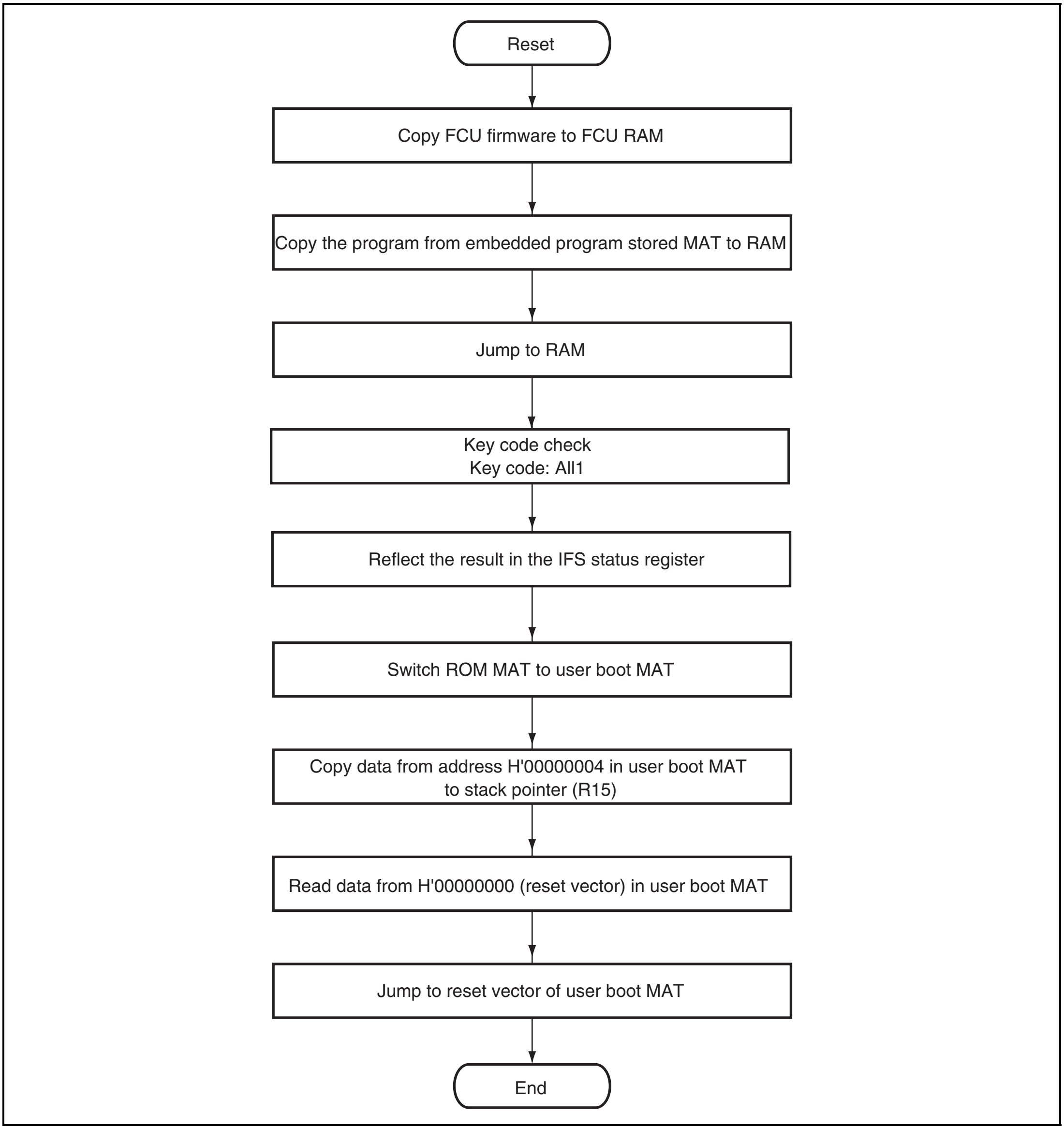

Reset
Copy FCU firmware to FCU RAM
Copy the program from embedded program stored MAT to RAM
Jump to RAM
Key code check
Key code: All1
Reflect the result in the IFS status register
Switch ROM MAT to user boot MAT
Copy data from address H'00000004 in user boot MAT
to stack pointer (R15)
Read data from H'00000000 (reset vector) in user boot MAT
Jump to reset vector of user boot MAT
End

Figure 26.30 Overview of Boot Sequence in User Boot Mode

26.7.2 User MAT Programming

The user MAT can be programmed by starting this LSI in user boot mode while the user MAT programming/erasing routine created by the user is stored in the user boot MAT. Be sure to copy the user MAT programming/erasing routine to the RAM and execute it in the RAM. When a key code is set, perform the key code approval. For details of the approval method, refer to section 27, Intelligent Flash Security (IFS). The user boot MAT is selected in the initial state in user boot mode; be sure to switch the memory MAT to the user MAT before starting programming. If an FCU command for ROM programming or erasure is issued while the user boot MAT is selected, the FCU does not program or erase the ROM. Figure 26.31 shows an example of the user MAT programming procedure.

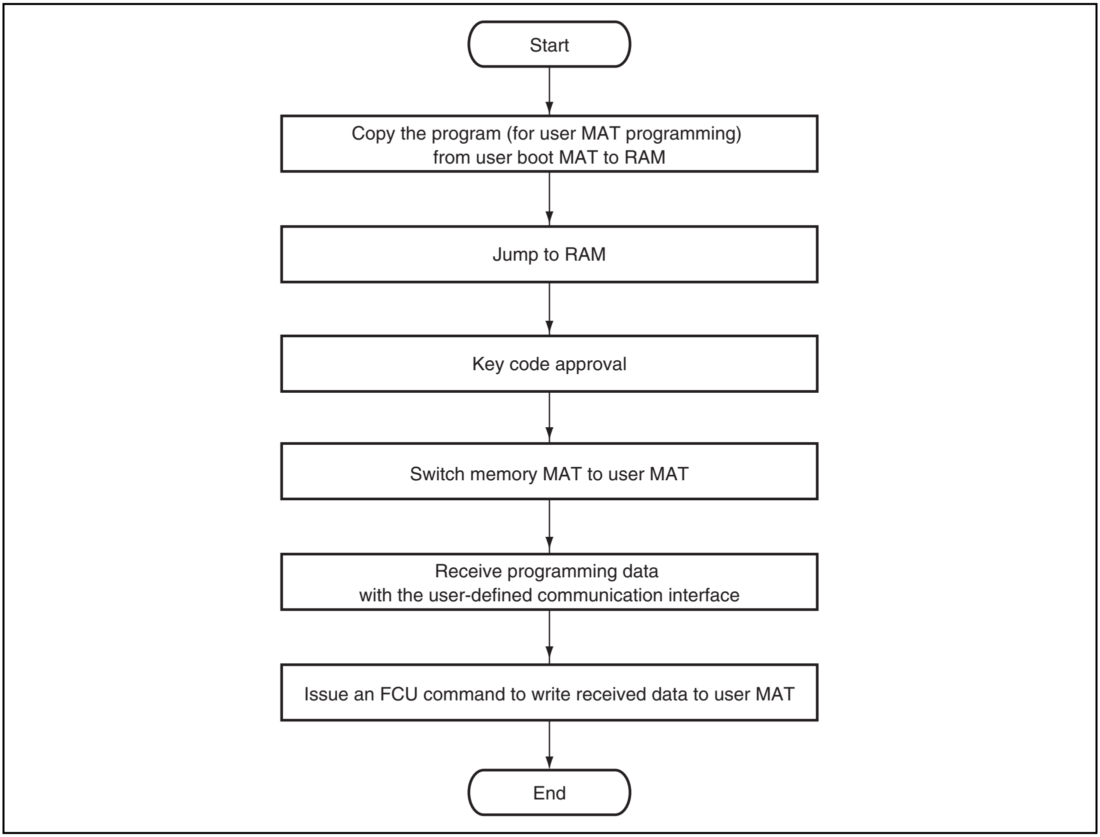

Start
Copy the program (for user MAT programming) from user boot MAT to RAM
Jump to RAM
Key code approval
Switch memory MAT to user MAT
Receive programming data with the user-defined communication interface
Issue an FCU command to write received data to user MAT
End

Figure 26.31 Example of User MAT Programming

26.8 Protection

There are three types of ROM programming/erasure protection: hardware, software, and error protection.

26.8.1 Hardware Protection

The hardware protection function disables ROM programming and erasure according to the LSI pin settings.

(1) Protection through FWE Pin

When a low level is applied to the FWE pin, the FWE bit in FPMON becomes 0. In this state, 11 cannot be written to the FENTRY6 to FENTRY0 bits in FENTRYR; that is, ROM P/E mode cannot be entered, which prevents the ROM from being programmed or erased.

When the FRDY bit is 1 and the FWE pin is driven low, the FCU clears the FENTRY6 to FENTRY0 bits to disable ROM programming and erasure. If the FRDY bit in FSTATR0 has already been set to 0 before the FWE pin is driven low, the FCU continues command processing. Even while processing a command, the FCU can accept a P/E suspend command. To resume programming or erasing the ROM, reset the FENTRY6 to FENTRY0 bits to the value that was set before being cleared, and then issue a P/E resume command.

If an attempt is made to issue a programming or erasing command to the ROM against the protection through the FWE pin, the FCU detects an error and enters command-locked state.

(2) Protection through Mode Pins

While the on-chip ROM is disabled, ROM programming, erasing, and reading are disabled. For the operating modes set through the mode pins of this LSI, refer to section 3, Operating Modes. In user boot mode or user program mode, the user boot MAT cannot be programmed or erased.

26.8.2 Software Protection

The software protection function disables ROM programming and erasure according to the control register settings or the lock bit settings in the user MAT. If an attempt is made to issue a programming or erasing command to the ROM against software protection, the FCU detects an error and enters command-locked state.

(1) Protection through FENTRYR

When the FENTRY6 bit in FENTRYR is 0, the 0.25-Mbyte ROM (read addresses: H'003C0000 to H'003FFFFF; program/erase addresses: H'80BC0000 to H'80BFFFFF) is set to ROM reading mode.
When the FENTRY5 bit in FENTRYR is 0, the 0.25-Mbyte ROM (read addresses: H'00380000 to H'003BFFFF; program/erase addresses: H'80B80000 to H'80BBFFFF) is set to ROM read mode.
When the FENTRY4 is 0, the 0.25-Mbyte ROM (read addresses: H'00340000 to H'0037FFFF; program/erase addresses: H'80B40000 to H'80B7FFFF) is set to ROM read mode.

When the FENTRY3 bit is 0, the 0.25-Mbyte ROM (read addresses: H'00300000 to H'0033FFFF; program/erase addresses: H'80B00000 to H'80B3FFFF) is set to ROM read mode.

When the FENTRY2 bit is 0, the 1-Mbyte ROM (read addresses: H'00200000 to H'002FFFFF; program/erase addresses: H'80A00000 to H'80AFFFFF) is set to ROM read mode. When the FENTRY1 bit is 0, the 1-Mbyte ROM (read addresses: H'00100000 to H'001FFFFF; program/erase addresses: H'80900000 to H'809FFFFF) is set to ROM read mode. When the FENTRY0 bit is 0, the 1-Mbyte ROM (read addresses: H'00000000 to H'000FFFFF; program/erase addresses: H'80800000 to H'808FFFFF) is set to ROM read mode. In ROM read mode, the FCU does not accept commands, so ROM programming and erasure are disabled. If an attempt is made to issue an FCU command in ROM read mode, the FCU detects an illegal command error and enters command-locked state (see section 26.8.3, Error Protection).

(2) Protection through Lock Bits

Each erasure block in the user MAT has a lock bit. When the FPROTCN bit in FPROTR is 0, the erasure block whose lock bit is set to 0 cannot be programmed or erased. To program or erase the erasure block whose lock bit is 0, set the FPROTCN bit to 1. If an attempt is made to issue a programming or erasing command against protection by lock bits, the FCU detects an programming/erasure error and enters command-locked state (see section 26.8.3, Error Protection).

26.8.3 Error Protection

The error protection function detects an illegal FCU command issued, an illegal access, or an FCU malfunction, and disables FCU command acceptance (command-locked state). While the FCU is in command-locked state, the ROM cannot be programmed or erased. To cancel command-locked state, issue a status register clear command while FASTAT is H'10.

While the CMDLKIE bit in FAEINT is 1, a flash interface error (FIFE) interrupt is generated if the FCU enters command-locked state (the CMDLK bit in FASTAT becomes 1). While the ROMAEINT bit in FAEINT is 1, an FIFE interrupt is generated if the ROMAE bit in FASTAT becomes 1.

Table 26.14 shows the error protection types dedicated for the ROM, those used in common by the ROM and the EEPROM, and the status bit values (the ILGLERR, ERSERR, and PRGERR bits in FSTATR0, the FCUERR, FRDTCT, and FRCRCT bits in FSTATR1, and the ROMAE bit in FASTAT) after each error detection. If the FCU enters command-locked state due to a command other than a suspend command issued during programming or erasure processing, the FCU continues programming or erasing the ROM. In this state, the P/E suspend command cannot suspend programming or erasure. If a command is issued in command-locked state, the ILGLERR bit becomes 1 and the other bits retain the values set due to the previous error detection.

Table 26.14 Error Protection Types

Error	Description	ILGLERR	ERSERR	PRGERR	FCUERR	FRDTCT	FRCRCT	ROMAE
FENTRYR setting error	The value set in FENTRYR is not H'0001, H'0002, H'0004, H'0008, H'0010, H'0020, or H'0080.	1	0	0	0	0	0	0
	The FENTRYR setting for resuming operation does not match that for suspending operation.	1	0	0	0	0	0	0

Error	Description	ILGLERR	ERSERR	PRGERR	FCUERR	FRDTCT	FRCRCT	ROMAE
Illegal command error	An undefined code has been specified in the first cycle of an FCU command.	1	0	0	0	0	0	0
	The value specified in the last of the multiple cycles of an FCU command is not H'D0.	1	0	0	0	0	0	0
	The command issued during programming or erasure is not a suspend command.	1	0	0	0	0	0	0
	A suspend command has been issued during operation that is neither programming nor erasure.	1	0	0	0	0	0	0
	A suspend command has been issued in suspended state.	1	0	0	0	0	0	0
	A resume command has been issued in a state that is not a suspended state.	1	0	0	0	0	0	0
	A programming or erasing command (program, lock bit program, block erase) has been issued in programming-suspended state.	1	0	0	0	0	0	0
	A block erase command has been issued in erasure-suspended state.	1	0	0	0	0	0	0
	A program, lock bit program, or non-interleaved program command has been issued for an erasure-suspended area in erasure-suspended state.	1	0	0	0	0	0	0
	The value specified in the second cycle of a program command is not H'80.	1	0	0	0	0	0	0
	A command has been issued in command-locked state.	1	0/1	0/1	0/1	0/1	0/1	0/1
Erasure error	An error has occurred during erasure processing.	0	1	0	0	0	0	0
	A block erase command has been issued for the erasure block whose lock bit is set to 0 while the FPROTCN bit in FPROTR is 0.	0	1	0	0	0	0	0
Programming error	An error has occurred during programming processing.	0	0	1	0	0	0	0
	A program, lock bit program, or program command has been issued for the erasure block whose lock bit is set to 0 while the FPROTCN bit in FPROTR is 0.	0	0	1	0	0	0	0

Error	Description	ILGLERR	ERSERR	PRGERR	FCUERR	FRDTCT	FRCRCT	ROMAE
FCU error	An error has occurred during CPU processing in the FCU.	0	0	0	1	0	0	0
FCU RAM ECC error	An ECC 1-bit error has been corrected during FCU RAM reading.	0	0	0	0	0	1	0
	An ECC 2-bit has been detected during FCU RAM reading.	0	0	0	1	1	0	0
ROM access error	A read access command has been issued to addresses H'80BC0000 to H'80BFFFFF while FENTRY6 = 1 in ROM P/E normal mode.	1	0	0	0	0	0	1
	A read access command has been issued to addresses H'80B80000 to H'80BBFFFF while FENTRY5 = 1 in ROM P/E normal mode.	1	0	0	0	0	0	1
	A read access command has been issued to addresses H'80B40000 to H'80B7FFFF while FENTRY4 = 1 in ROM P/E normal mode.	1	0	0	0	0	0	1
	A read access command has been issued to addresses H'80B00000 to H'80B3FFFF while FENTRY3 = 1 in ROM P/E normal mode.	1	0	0	0	0	0	1
	A read access command has been issued to addresses H'80A00000 to H'80AFFFFF while FENTRY2 = 1 in ROM P/E normal mode.	1	0	0	0	0	0	1
	A read access command has been issued to addresses H'80900000 to H'809FFFFF while FENTRY1 = 1 in ROM P/E normal mode.	1	0	0	0	0	0	1
	A read access command has been issued to addresses H'80800000 to H'808FFFFF while FENTRY0 = 1 in ROM P/E normal mode.	1	0	0	0	0	0	1
	An access command has been issued to addresses H'80BC0000 to H'80BFFFFF while FENTRY6 = 0	1	0	0	0	0	0	1
	An access command has been issued to addresses H'80B80000 to H'80BBFFFF while FENTRY5 = 0	1	0	0	0	0	0	1
	An access command has been issued to addresses H'80B40000 to H'80B7FFFF while FENTRY4 = 0	1	0	0	0	0	0	1

Error	Description	ILGLERR	ERSERR	PRGERR	FCUERR	FRDTCT	FRCRCT	ROMAE
ROM access error	An access command has been issued to addresses H'80B00000 to H'80B3FFFF while FENTRY3 = 0	1	0	0	0	0	0	1
	An access command has been issued to addresses H'80A00000 to H'80AFFFFF while FENTRY2 = 0	1	0	0	0	0	0	1
	An access command has been issued to addresses H'80900000 to H'809FFFFF while FENTRY1 = 0	1	0	0	0	0	0	1
	An access command has been issued to addresses H'80800000 to H'808FFFFF while FENTRY0 = 0	1	0	0	0	0	0	1
	A read access command has been issued to addresses H'00000000 to H'003FFFFF while the FENTRYR register value is not H'0000.	1	0	0	0	0	0	1
	A ROM programming or erasing command (interleaved program, lock bit program, or block erase command) has been issued while the user boot MAT is selected.	1	0	0	0	0	0	1
	An access command has been issued to an address other than the addresses for ROM programming/erasure H'80800000 to H'80807FFF while the user boot MAT is selected.	1	0	0	0	0	0	1

Notes: 1. When the FRDCLE bit in FRAMECCR is set to 1.
2. When the FRCCLE bit in FRAMECCR is set to 1.

26.9 Usage Notes

26.9.1 Switching between User MAT and User Boot MAT

The user MAT and user boot MAT are allocated to the same address area. If the ROM area is accessed during switching between the user MAT and user boot MAT, an unexpected MAT may be accessed because the number of cycles required to access the ROM area depends on the internal bus status. When the ROM cache function is enabled, the previously stored data is left in the ROM cache even after MAT switching; note that a cache hit may occur when a newly selected MAT is accessed at the same address as the data stored in the cache. To avoid such unexpected behavior, take the following steps before and after MAT switching.

1. Modifying interrupt settings before MAT switching

   There are two ways to avoid ROM area access due to an interrupt during MAT switching: one is to specify the interrupt vector fetch destination outside the ROM area through the vector base register (VBR) setting in the CPU, and the other is to mask interrupts. Note that NMI interrupts cannot be masked in this LSI; when masking interrupts to avoid ROM area access in this LSI, design the system so that no NMI is generated during MAT switching.

2. Switching between MATs through a program outside the ROM area

   To avoid CPU instruction fetch in the ROM area during MAT switching, execute the MAT switching processing outside the ROM area.

3. Performing dummy read of ROMMAT

   After writing to ROMMAT to switch between MATs, perform a dummy read of ROMMAT to ensure that the register write is completed.

4. Flushing the ROM cache during MAT switching

   Flush all lines of the ROM cache by writing 1 to the RCF bit in RCCR (see section 29, ROM Cache (ROMC)).

5. Executing five or more NOP instructions

   After the dummy read of ROMMAT, execute five or more NOP instructions.

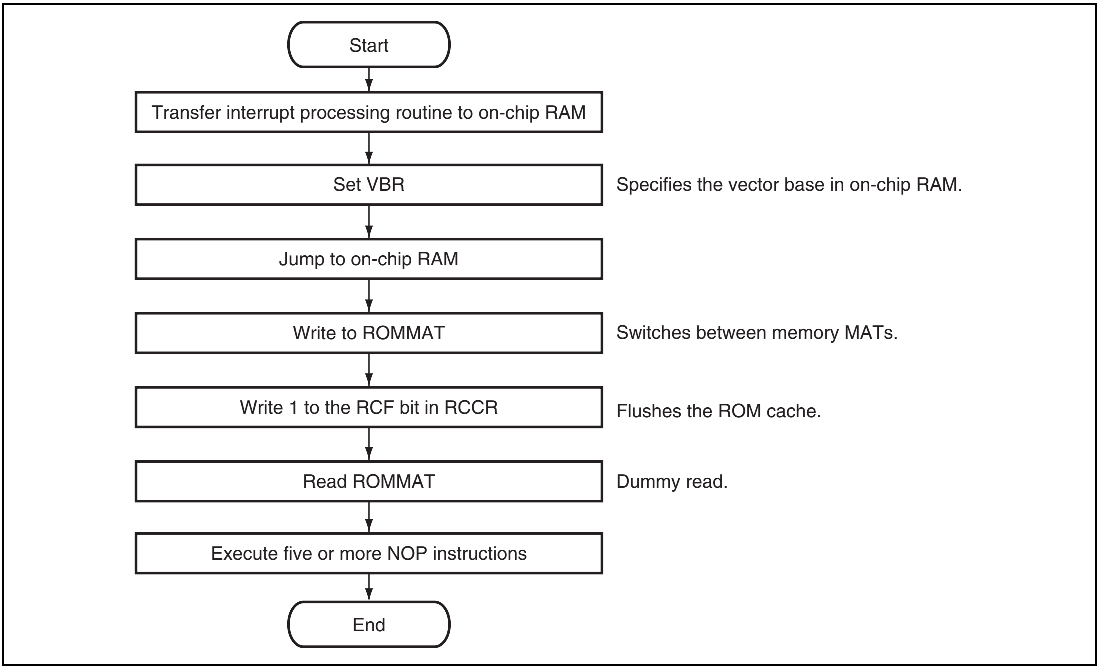

Start
Transfer interrupt processing routine to on-chip RAM
Set VBR
Specifies the vector base in on-chip RAM.
Jump to on-chip RAM
Write to ROMMAT
Switches between memory MATs.
Write 1 to the RCF bit in RCCR
Flushes the ROM cache.
Read ROMMAT
Dummy read.
Execute five or more NOP instructions
End

Figure 26.32 Example of MAT Switching Steps

26.9.2 Other Notes

(1) State in which AUD Operation Is Disabled and Interrupts Are Ignored

In the following modes or period, the AUD is in module standby mode and cannot operate. The NMI or maskable interrupt requests are ignored.

- Boot mode
- The program in the embedded program stored MAT is being executed immediately after the LSI is started in user boot mode

(2) Programming-/Erasure-Suspended Area

The data stored in the programming-suspended or erasure-suspended area is undetermined. To avoid malfunction due to undefined read data, ensure that no instruction is executed or no data is read from the programming-suspended or erasure-suspended area.

To avoid instruction fetch from the programming-suspended or erasure-suspended area, which may be caused by prefetch by the ROM cache, ensure that no instruction is fetched within 32 bytes from the start address of the programming-suspended or erasure-suspended area.

During ROM cache prefetch, the destination of a branch instruction is also accessed. If the destination can be in the programming-suspended or erasure-suspended area, disable the prefetch function of the ROM cache.

(3) Compatibility with Programming/Erasing Program of Conventional F-ZTAT SH Microcomputers

The flash memory programming/erasing program used for conventional F-ZTAT SH microcontrollers does not work with this LSI.

(4) FWE Pin State

Ensure that the FWE pin level does not change during programming or erasure. Even if the FWE level goes low, the current programming or erasure continues, but FENTRYR is cleared after the programming or erasure processing is completed. In this state, if an FCU command is issued without resetting FENTRYR, the FCU detects a ROM access error and enters command-locked state.

(5) Reset during Programming or Erasure

To reset the FCU by setting the FRESET bit in the FRESETR register during programming or erasure, hold the FCU in the reset state for a period of t_{RESW2} (see section 34, Electrical Characteristics). Since a high voltage is applied to the ROM during programming and erasure, the FCU has to be held in the reset state long enough to ensure that the voltage applied to the memory unit has dropped. Do not read from the ROM while the FCU is in the reset state.

When a power-on reset is generated by asserting the $\overline{\text{RES}}$ pin during programming or erasure of the flash memory, hold the reset state for a period of t_{RESW2} (see section 34, Electrical Characteristics). In a power-on reset, not only does the voltage applied to the memory unit have to drop, but the power supply for the ROM and its internal circuitry also have to be initialized. Thus, the reset state must be maintained over a longer period than in the case of resetting the FCU.

While programming or erasure is performed, do not generate an internal reset caused by WDT counter overflow. A reset caused by WDT cannot ensure a sufficient time required for voltage drop for the memory unit, initialization of the power supply for the ROM, or initialization of its internal circuit.

When either a power-on reset by asserting the $\overline{\text{RES}}$ pin, or an FCU reset by setting the FRESET bit in the FRESETR register, is executed during programming or erasure, the whole data in the programming or erasure area becomes undefined.

(6) Prohibition of Additional Programming

One area cannot be programmed twice in succession. To program an area that has already been programmed, be sure to erase the area before reprogramming.

(7) Suspension by Programming or Erasure Suspension

When programming or erasure is suspended by a programming or an erasure suspend command, the programming or erasure must be completed by a resume command.

(8) Power off during Programming or Erasure

Do not switch the power off during programming or erasure. As a high voltage is applied to the ROM during programming and erasure, a certain period is required for this voltage to fall.

Thus, to allow for cases where switching the power off during programming or erasure cannot be avoided, design the system so that at least the V_{CC} holding time at PV_{CC} shutdown (t_{VCCH}) is secured by assertion of the $\overline{\text{HSTBY}}$ signal causing a transition to the hardware standby state if the power is switched off. For details, see section 34, Electrical Characteristics.

(9) Prohibition of Clearing FRDCLE Bit to 0

Whenever the FRDTCT bit in FSTATR1 is set to 1, the FCU has to enter command-locked state, because the FCU command operation cannot be guaranteed. Thus, do not clear the FRDCLE bit in FRAMECCR to 0.

(10) Note on Transition to ROM Read Mode

When transiting to ROM read mode by changing FENTRY6 to FENTRY0 bit in FENTRYR register from 1 to 0, write 0 to FENTRY6 to FENTRY0 bit and execute a minimum of 5 NOP instructions after performing a dummy read of FENTRYR register.

Section 27 Intelligent Flash Security (IFS)

This LSI provides the intelligent flash security (IFS) function that protects the on-chip ROM against programming or erasure.

27.1 Features

- The IFS function protects the on-chip ROM by prohibiting programming or erasure.
- The IFS function provides three types of security setting state.

  Transition between these states is made by key code set, reset, or approval.

  Unprotected state: No key code is set and the IFS function does not protect the on-chip ROM against programming or erasure.

  Protected-unlocked state: A key code is set but the IFS function does not protect the on-chip ROM against programming or erasure.

  Protected-locked state: A key code is set and the IFS function protects the on-chip ROM against programming or erasure.
- The IFS function provides two levels of on-chip ROM protection against programming or erasure.

  Security level 1: In ASE mode, the debugging interface through the H-UDI is enabled even in the protected state.

  Security level 2: In ASE mode, the debugging interface through the H-UDI is always disabled.
- Key codes are set, reset, and approved by flash sequencer commands.
- A key code is 64 bits long. Any value can be selected by the user.
- The on-chip ROM can also be protected against reading by appropriately setting the security state.
- The EEPROM can always be programmed and erased regardless of the security setting state.
- The key code values must be managed by the user applications. The key code written by an FCU command cannot be read directly from the register. The written key code is approved by a check inside the FCU initiated by an FCU command.

Note: This LSI has the ASE mode for debugging with an external emulator connected. When the ASEMD pin is set to 1, this LSI operates in the ASE mode. This LSI provides a function for connecting a debugger through the high-performance user debugging interface (H-UDI), which is available only in the ASE mode. The IFS restricts the debugger connection function through H-UDI in the ASE mode according to the security level.

If the ASE mode is specified without connecting an emulator, correct operation is not guaranteed. When operating this LSI without connecting an emulator, be sure to set the ASEMD pin to 0 to select the normal operation mode.

27.2 Register Descriptions

Table 27.1 shows the IFS-related registers. This section only describes the IFS-related functions. For the ROM-related functions, refer to section 26, ROM.

Table 27.1 IFS-Related Register Configuration

Register Name	Symbol	R/W*1	Address	Access Size
IFS control mode register	IFSCMR	R/(W)*2	H'FFFFA890	8, 16
IFS status register	IFSSR	R	H'FFFFA912	8, 16

Notes: 1. In on-chip ROM disabled mode, the IFS-related registers are always read as 0 and writing to them is ignored.

2. This register can be written to only when a specified value is written to the upper byte in word access. The data written to the upper byte is not stored in the register.

27.2.1 IFS Control Mode Register (IFSCMR)

The IFS control mode register is used to set the IFS control mode. Writing to this register in on-chip ROM disabled mode has no effect. The value read out is H'0000. This register cannot be changed during the key code setting, reset, or approval.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KEY								—	—	—	—	—	—	—	IFSCM
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R	R	R	R	R	R/W

Note: This register can be written to only when a specified value is written to the upper byte in word access. The data written to the upper byte is not stored in the register.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	KEY	All 0	R/(W)*	Key Code These bits enable or disable modification of the IFSCM bit. Data written to these bits are not stored.
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	IFSCM	0	R/W	IFS Control Mode Bit This bit should be set when setting, reset, and the key code approval is to be performed. 0: IFS control mode is not selected. 1: IFS control mode is selected. Condition for writing to be effective: Satisfaction of all of the following conditions • On-chip ROM is in enabled mode. • FRDY bit in the FSTATR0 is 1. • H'6E is written to the KEY bits at the same time through a word-access operation Set condition: • Writing of 1 to IFSCM with all conditions for writing to be effective met and the value in the FENTRYR register being H'0000 Conditions for clearing: • The FRDY bit in the FSTATR0 register being 1. • Attempting to write through byte access • Writing with a key value other than H'6E • Writing of 0 to the IFSCM register when all conditions for writing to be effective are met.

Note: * This register can be written to only when a specified value is written to the upper byte in word access. The data written to the upper byte is not stored in the register.

27.2.2 IFS Status Register (IFSSR)

The IFS register is used to monitor the security conditions set up by the IFS function. In on-chip ROM disabled mode, reading-out value has no effect.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	IFSST[2:0]		
Initial value:	0	0	0	0	0	0/1*	0/1*	0/1*
R/W:	R	R	R	R	R	R	R	R

Note: The initial value depends on the mode of operation and state of the key code. Refer to table 27.2.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	IFSST[2:0]	000/111*	R	IFS Status Bits These bits can be used to monitor the security conditions set up by the IFS function. 000: Unprotected state 001: Protected-unlocked state (security level 1) 010: Not used (not specified) 011: Protected-unlocked state (security level 2) 100: Not used (not specified) 101: Protected-locked state (security level 1) 110: Not used (not specified) 111: Protected-locked state (security level 2)

Note: * The initial value depends on the mode of operation and state of the key code. Refer to table 27.2.

Table 27.2 IFSST[2:0] Bit Initial Value

Operation mode	Key code setting mode	Security level for the set key code	IFSST [2:0] Initial Value	Security setting mode
ASE mode (ASEMD pin high level*[1])	Reset	None	000	Unprotect
	Set	Level 1	101	Protect lock (Level 1)
		Level 2	—	Protect lock (Level 2) (Debug connection invalid)
Product chip mode User boot mode (ASEMD pin low level)	Reset	None	000	Unprotect
	Set	Level 1	101	Protect lock (Level 1)
		Level 2	111	Protect lock (Level 2)
Product chip mode User program mode (ASEMD pin low level)	Reset	None	111*[1]	Protect lock (Level 2) *[2]
	Set	Level 1	111*[1]	Protect lock (Level 2) *[2]
		Level 2	111*[1]	Protect lock (Level 2) *[2]

Notes: *1 When the security level has been changed during operation, the initial value of IFSST bits after a user reset matches the previous security level. To make the initial value of IFSST bits match the new security level, assert an ASE reset.

*2 In the user program mode, key code setting conditions and the initial value of the IFSST [2:0] bits do not match. To make them match, key code approval by the user software is required.

27.3 Operation

27.3.1 Security Setting States and Operations

The IFS controls five types of security setting state.

According to the selected state, the IFS restricts ROM programming and erasure operations and the H-UDI function for connecting a debugger.

To protect the on-chip ROM against programming or erasure or to disable the H-UDI interface, set, reset, or approve a key code as appropriate in the IFS.

Table 27.3 Security Setting States

State Name	On-Chip ROM Programming or Erasure	Key Code	H-UDI Interface
Unprotected	Not protected	Reset	Not affected
Protected-unlocked (security level 1)	Not protected	Set	Not affected
Protected-locked (security level 1)	Protected	Set	Not affected
Protected-unlocked (security level 2)	Not protected	Set	Not affected
Protected-locked (security level 2)	Protected	Set	Disabled

The following describes the operation in each security setting state.

(1) Unprotected State

No key code is set. In this state, the IFS function does not protect the on-chip ROM against programming or erasure. The H-UDI interface function in the ASE mode is not affected by the IFS function and a debugger can be connected.

(2) Protected-Unlocked State (Security Level 1)

The key code for level 1 is set. In this state, although a key code is set, the on-chip ROM protection by the IFS function against programming or erasure is disabled. The H-UDI interface function in the ASE mode is not affected by the IFS function and a debugger can be connected.

(3) Protected-Locked State (Security Level 1)

The key code for level 1 is set and the protection function is enabled. In this state, the IFS function protects the on-chip ROM by invalidating programming and erasure. The H-UDI interface function in the ASE mode is not affected by the IFS function and a debugger can be connected.

In this state, correct values cannot be read from the on-chip ROM once an instruction is fetched from the external address space (CS0 to CS3 areas).

(4) Protected-Unlocked State (Security Level 2)

The key code for level 2 is set. In this state, although a key code is set, the on-chip ROM protection by the IFS function against programming or erasure is disabled. The H-UDI interface function in the ASE mode is not affected by the IFS function and a debugger can be connected.

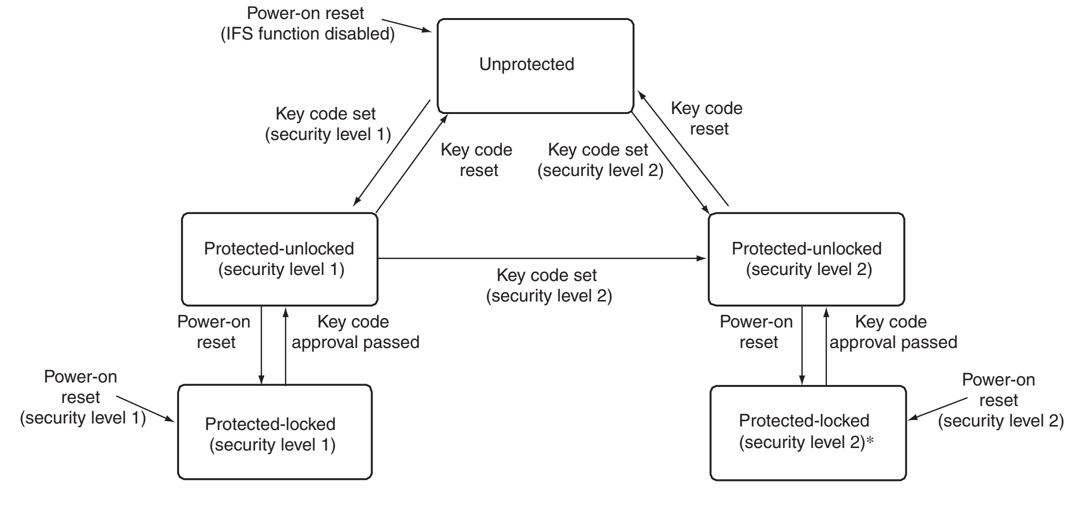
(5) Protected-Locked State (Security Level 2)

The key code for level 2 is set and the protection function is enabled. In this state, the IFS function protects the on-chip ROM by invalidating programming and erasure. The IFS function also restricts the H-UDI interface function in the ASE mode and no debugger can be connected.

In this state, correct values cannot be read from the on-chip ROM once an instruction is fetched from the external address space (CS0 to CS3 areas).

27.3.2 Security Setting State Transition

Figure 27.1 shows the security setting state transition. The figure is for the operation modes in which programming/erasure is enabled.



Note: * In the user program mode, this LSI is started in the protected-locked state (security level 2).

Figure 27.1 IFS Security Setting State Transition

The following describes each transition between security setting states.

(1) Transition from Unprotected State

When a key code is set, transition is made to the protected-unlocked state (security level 1) or protected-unlocked state (security level 2) according to the level of the key code.

(2) Transition from Protected-Unlocked State (Security Level 1)

When the key code is reset, transition is made to the unprotected state.

At a power-on reset in the protected-unlocked state (security level 1), transition is made to the protected-locked state (security level 1).

When a key code (security level 2) is set, transition is made to the protected-unlocked state (security level 2).

(3) Transition from Protected-Locked State (Security Level 1)

When the key code approval check is done and passed, transition is made to the protected-unlocked state (security level 1).

(4) Transition from Protected-Unlocked State (Security Level 2)

When the key code is reset, transition is made to the unprotected state.

At a power-on reset in the protected-unlocked state (security level 2), transition is made to the protected-locked state (security level 2).

(5) Transition from Protected-Locked State (Security Level 2)

When the key code approval check is done and passed, transition is made to the protected-unlocked state (security level 2).

27.3.3 IFS Control Mode

To set, reset, or approve a key code, transition should be made from the ROM read mode to the IFS control mode.

To enter the IFS control mode, set the KEY bits to H'6E and the IFSCM bit to 1 in the IFS control mode register (IFSCMR).

Figure 27.2 shows mode transition.

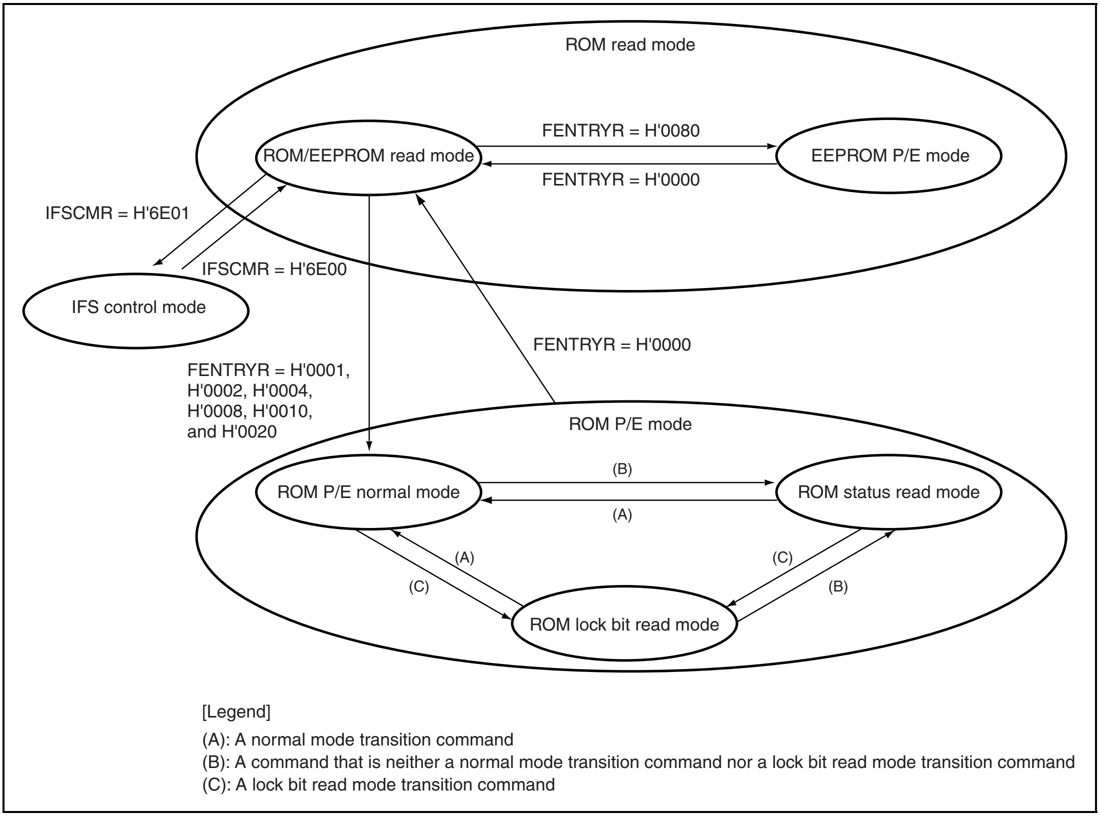

ROM read mode
ROM/EEPROM read mode
FENTRYR = H'0080
FENTRYR = H'0000
EEPROM P/E mode
IFSCMR = H'6E01
IFSCMR = H'6E00
IFS control mode
FENTRYR = H'0000
FENTRYR = H'0001, H'0002, H'0004, H'0008, H'0010, and H'0020
ROM P/E mode
ROM P/E normal mode
(B)
(A)
ROM status read mode
(A)
(C)
(C)
(B)
ROM lock bit read mode
[Legend]
(A): A normal mode transition command
(B): A command that is neither a normal mode transition command nor a lock bit read mode transition command
(C): A lock bit read mode transition command

Figure 27.2 Mode Transition

For mode specifications other than those of the IFS control mode, refer to section 26, ROM.

27.3.4 Key Code Set, Reset, and Approval

Use FCU commands to manipulate the key code for setting the security in the IFS function. There are three FCU commands related to the IFS function as listed in table 27.4. The FCU commands are valid only in the IFS control mode. The FCU state must be checked through the flash status register 0 (FSTATR0) and flash status register 1 (FSTATR1).

Table 27.4 FCU Commands Related to the IFS Functions

Command	Operation
Key code set	Sets the key code
Key code reset	Resets the key code
Key code approval	Checks if the key code matches the pre-specified one.

Table 27.5 shows the FCU command format. Table 27.6 shows the acceptable commands in the IFS control mode. When a command that cannot be accepted is issued, the FCU enters the command-locked state.

Table 27.5 FCU Command Format

Item		First Cycle		Second Cycle		Third Cycle		Fourth to Sixth Cycles		Seventh Cycle	
Command	Number of Bus Cycles	Address	Data	Address	Data	Address	Data	Address	Data	Address	Data
Key code set	7	RA	H'E8	RA	H'04	WA	WD_1	RA	WD_n	RA	H'D0
Key code reset	2	RA	H'20	BA	H'D0	—	—	—	—	—	—
Key code approval	7	RA	H'99	RA	H'04	WA	WD_1	RA	WD_n	RA	H'D0
Status register clear	1	RA	H'50	—	—	—	—	—	—	—	—

[Legend]

- Columns of address

  RA: FCU command issuance address in the IFS control mode
  When IFSCM is 1: An address in the range from H'807FE000 to H'807FFFFF

  WA: Key code set / approval address
  When IFSCM is 1:

  - Key code set

    Security level 1: H'807FFF00
    Security level 2: H'807FFF08

  - Key code approval: H'807FFF08

  BA: Key code reset address
  When IFSCM is 1: H'807FFF00

- Columns of data

  WD_n: n-th word of key code (n = 1 to 4)

Table 27.6 FCU Modes/States and Acceptable Commands (IFS Control Mode)

	IFS Control Mode			
Item	**Key Code Set/ Reset Processing**	**Command-Locked State**	**Protected-Locked State**	**Other State**
FRDY bit in FSTATR0	0	0/1	1	1
CMDLK bit in FASTAT	0	1	0	0
IFSST bits in IFSSR	Other than 101/111	*2	101/111	Other than 101/111
Key code set	×	×	×	A
Key code reset	×	×	×	A
Status register clear	×	A	A	A
Key code approval	×	×	A	A

[Legend]

A: Acceptable

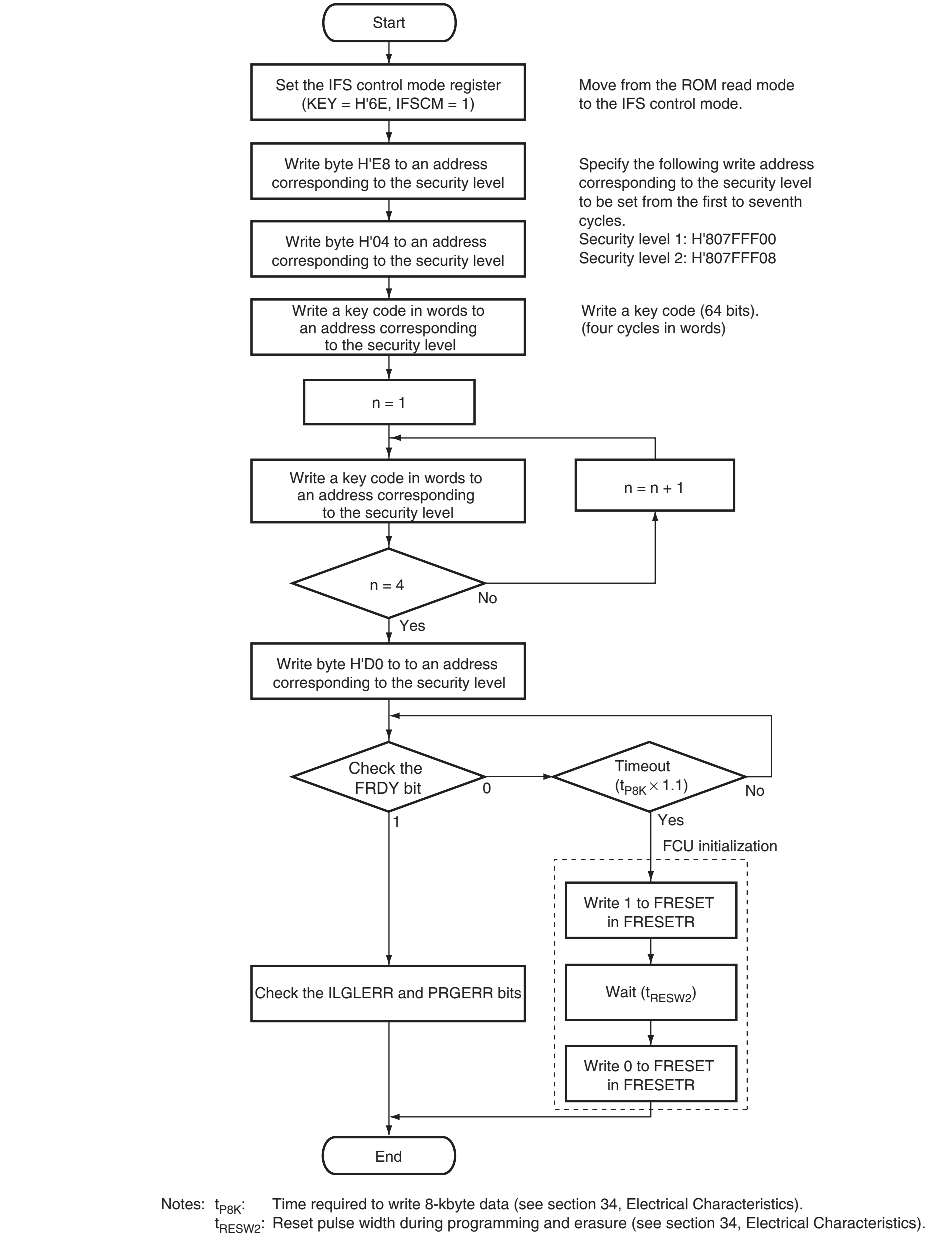
×: Not acceptable

Notes: *1 Including unprotected state / protected-unlocked state

*2 Not depending on the state of the IFSST bits

The following shows the procedures for key code set, reset, and approval.

(1) Key Code Set Procedure


Start
Set the IFS control mode register (KEY = H'6E, IFSCM = 1)
Move from the ROM read mode to the IFS control mode.
Write byte H'E8 to an address corresponding to the security level
Specify the following write address corresponding to the security level to be set from the first to seventh cycles.
Security level 1: H'807FFF00
Security level 2: H'807FFF08
Write byte H'04 to an address corresponding to the security level
Write a key code in words to an address corresponding to the security level
Write a key code (64 bits). (four cycles in words)
n = 1
Write a key code in words to an address corresponding to the security level
n = n + 1
n = 4
No
Yes
Write byte H'D0 to to an address corresponding to the security level
Check the FRDY bit
0
1
Timeout (t_{P8K} × 1.1)
No
Yes
FCU initialization
Write 1 to FRESET in FRESETR
Wait (t_{RESW2})
Write 0 to FRESET in FRESETR
Check the ILGLERR and PRGERR bits
End

Notes: t_{P8K}: Time required to write 8-kbyte data (see section 34, Electrical Characteristics).
t_{RESW2}: Reset pulse width during programming and erasure (see section 34, Electrical Characteristics).

Figure 27.3 Key Code Set Procedure

To set a key code, use the key code set command in the FCU.

Specify a write address from the first to seventh cycles of the command according to the security level to be set; specify H'807FFF00 for security level 1, or H'807FFF08 for security level 2.

Set the IFSCM bit in IFSCMR for a transition from the ROM read mode to the IFS control mode. In the IFS control mode, write H'E8 in the first cycle of the key code set command and H'04 in the second cycle, in bytes, to the address corresponding to the security level. Access the peripheral bus in words from the third to sixth cycles of the command.

Write a 64-bit key code in words from the third to sixth cycles of the command.

Write H'D0 in bytes to the address corresponding to the security level in the seventh cycle; the FCU will then start the key code set processing. Read the FRDY bit in FSTATR0 to confirm that the key code set processing is completed.

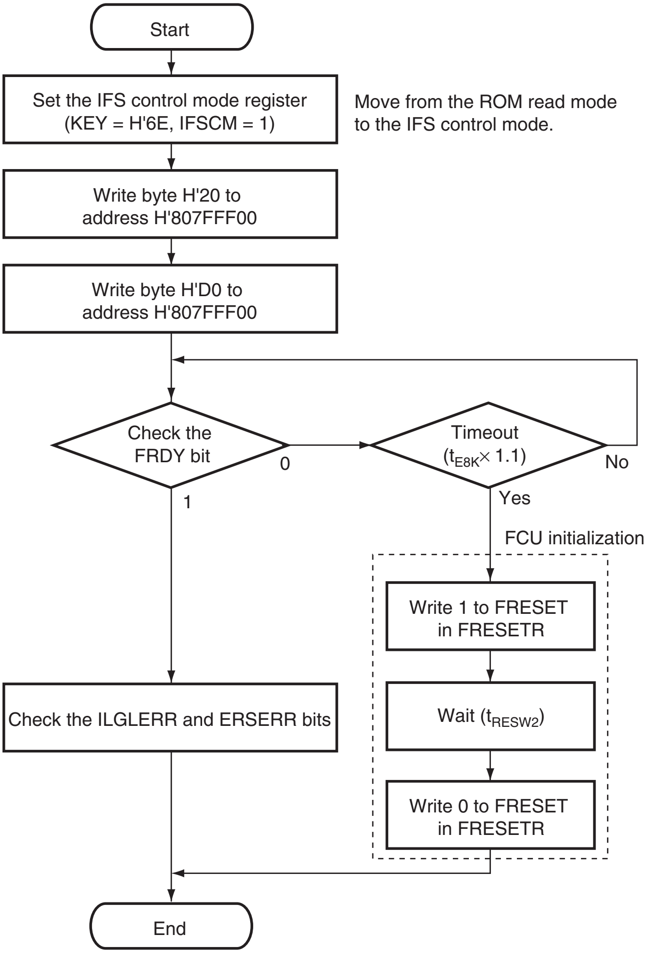
If a key code set command is issued while an illegal combination of addresses is specified, the FCU detects an error and enters command-locked state. To cancel command-locked state, issue a status register clear command while FASTAT is H'10. For FASTAT and the status register clear command, refer to section 26, ROM.

Note on Key Code Set (Additional Writing Processing):

While a key code is set, another key code cannot be set. Note that only when security level 2 is specified while a key code for security level 1 is set, another key code can be set as soon as transition is made to security level 2.

Before setting a key code, be sure to reset the previous key code.

(2) Key Code Reset Procedure


Start
Set the IFS control mode register (KEY = H'6E, IFSCM = 1)
Move from the ROM read mode to the IFS control mode.
Write byte H'20 to address H'807FFF00
Write byte H'D0 to address H'807FFF00
Check the FRDY bit
0
1
Timeout (t_{E8K}× 1.1)
No
Yes
FCU initialization
Write 1 to FRESET in FRESETR
Wait (t_{RESW2})
Write 0 to FRESET in FRESETR
Check the ILGLERR and ERSERR bits
End

Notes: t_{E8K}: Time required to erase a 8-kbyte block (see section 34, Electrical Characteristics).
t_{RESW2}: Reset pulse width during programming and erasure (see section 34, Electrical Characteristics).

Figure 27.4 Key Code Reset Procedure

To reset a key code, use the key code reset command in the FCU.

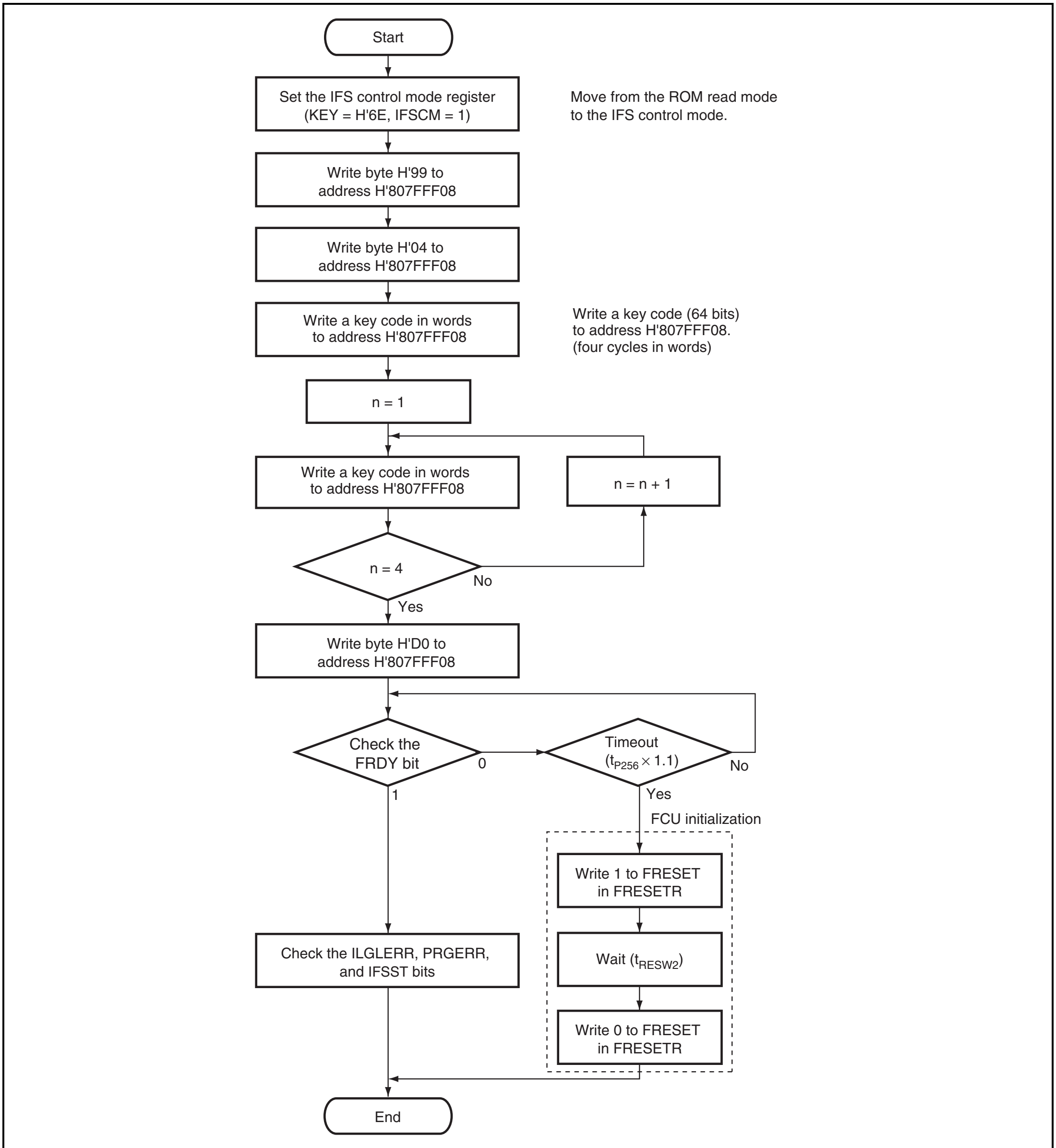
Set the IFSCM bit in IFSCMR for a transition from the ROM read mode to the IFS control mode. In the IFS control mode, write H'20 in bytes to address H'807FFF00 in the first cycle of the key code reset command and write H'D0 in bytes to address H'807FFF00 in the second cycle; the FCU will then start the key code reset processing. Read the FRDY bit in FSTATR0 to confirm that the key code reset processing is completed.

If a key code reset command is issued while an illegal combination of addresses is specified, the FCU detects an error and enters command-locked state. To cancel command-locked state, issue a status register clear command while FASTAT is H'10. For FASTAT and the status register clear command, refer to section 26, ROM.

Note on Key Code Reset:

The key code reset processing is done for both key codes for security levels 1 and 2. While key codes are set for both security levels 1 and 2, it is not possible to reset only the key code for security level 2.

(3) Key Code Approval Procedure


Start
Set the IFS control mode register (KEY = H'6E, IFSCM = 1)
Move from the ROM read mode to the IFS control mode.
Write byte H'99 to address H'807FFF08
Write byte H'04 to address H'807FFF08
Write a key code in words to address H'807FFF08
Write a key code (64 bits) to address H'807FFF08. (four cycles in words)
n = 1
Write a key code in words to address H'807FFF08
n = n + 1
n = 4
No
Yes
Write byte H'D0 to address H'807FFF08
Check the FRDY bit
0
1
Timeout (t_{P256} × 1.1)
No
Yes
FCU initialization
Write 1 to FRESET in FRESETR
Wait (t_{RESW2})
Write 0 to FRESET in FRESETR
Check the ILGLERR, PRGERR, and IFSST bits
End

Notes: t_{P256}: Time required to write 256-byte data (see section 34, Electrical Characteristics).
t_{RESW2}: Reset pulse width during programming and erasure (see section 34, Electrical Characteristics).

Figure 27.5 Key Code Approval Procedure

To approve a key code, use the key code approval command in the FCU. Specify H'807FFF08 for the address from the first to seventh cycles of the key code approval command.

Set the IFSCM bit in IFSCMR for a transition from the ROM read mode to the IFS control mode. In the IFS control mode, write H'99 in the first cycle of the key code approval command and H'04 in the second cycle, in bytes. Write a 64-bit key code in words from the third to sixth cycles of the command.

Write H'D0 in bytes in the seventh cycle; the FCU will then start the key code approval processing. The protection status and security level as a result of the approval is reflected in IFSST bits in IFSSR; read these bits to confirm that the key code approval processing is completed.

If a key code approval command is issued while an illegal combination of addresses is specified, the FCU detects an error and enters command-locked state. To cancel command-locked state, issue a status register clear command while FASTAT is H'10. For FASTAT and the status register clear command, refer to section 26, ROM.

Note on Key Code Approval:

The key code approval processing is done for both key codes for security levels 1 and 2. If key code approval for security level 1 is passed while key codes are set for both security levels 1 and 2, security level 2 takes priority and transition is not made to the protected-unlocked state (security level 1).

If a timeout occurs during approval command execution, the key code is not approved.

27.3.5 Boot Sequence in Each Operating Mode

- Boot Mode

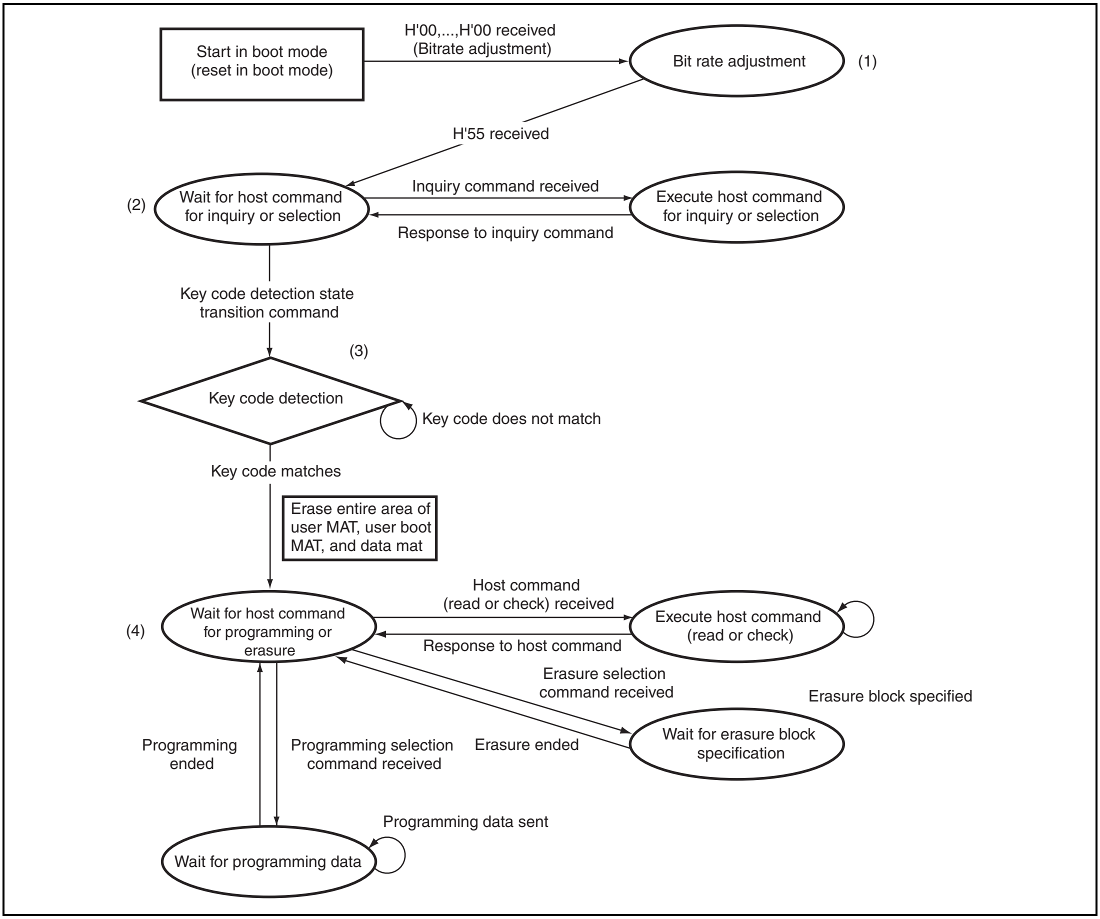


Figure 27.6 Boot Sequence in Boot Mode

(1) Bit Rate Adjustment (Same description as section 26.5.2, State Transition in Boot Mode)

After this LSI is started in boot mode, it automatically adjusts the bit rate for communications between the host and SCI_A. After automatically adjusting the bit rate, the LSI sends H'00 to the host. After the LSI has successfully received H'55 from the host, the LSI waits for a host command for inquiry or selection. For details on bit rate adjustment, see section 26.5.3, Automatic Adjustment of Bit Rate.

(2) Waiting for Host Command for Inquiry or Selection

In this state, the host inquires about the size of the MATs, structure of the MATs, addresses where the MATs start, state of support, the key code, etc., and selects the device, clock mode, and bit rate. Upon reception of a key code detection command from the host, this LSI performs a key code request. Upon reception of a key code detection state transition command sent from the host, this LSI enters the key code detection state.

(3) Key Code Detection

In this state, this LSI checks whether the key code sent by the host matches the one stored in this LSI. When the code matches, this LSI erases the entire area of each of the user MAT, user boot MAT, and data MAT, and enters the programming/erasure command wait state.

Note on Boot Mode Operation (Operation on Approval Failure):

When the key code approval has failed, the IFS returns a key code error to the host and waits for a new key code to be entered.

(4) Waiting for Host Command for Programming or Erasure

In this state, this LSI performs programming or erasure in accord with commands from the host. In addition, the LSI performs key code set or reset, also in this state. Depending on the received command, the LSI will enter the programming data wait state, erasure block specification wait state, or command (read or check) processing state.

Upon reception of a programming selection command, the LSI waits for data to be programmed. After the programming selection command, the host sends the address where programming is to start and the data for programming. Specifying H'FFFFFFFF as the start address for programming terminates programming processing and the LSI makes a transition from the programming data wait state to the programming/erasure command wait state.

Upon reception of an erasure selection command, the LSI waits for erasure block specification. After the erasure selection command, the host sends the erasure-block number. Specifying H'FF as the erasure block number terminates erasure processing and the LSI makes a transition from the erasure block specification wait state to programming/erasure command wait state.

In addition to programming and erasing commands, many other host commands are provided for use in the programming/erasure command wait state; these include commands for sum checking, blank checking (erasure checking), memory reading, key code setting and resetting, and inquiry on the state of the user MAT and user-boot MAT. For details on these host commands, see section 26.5.5, Programming/Erasing Host Command Wait State.

- User Boot Mode

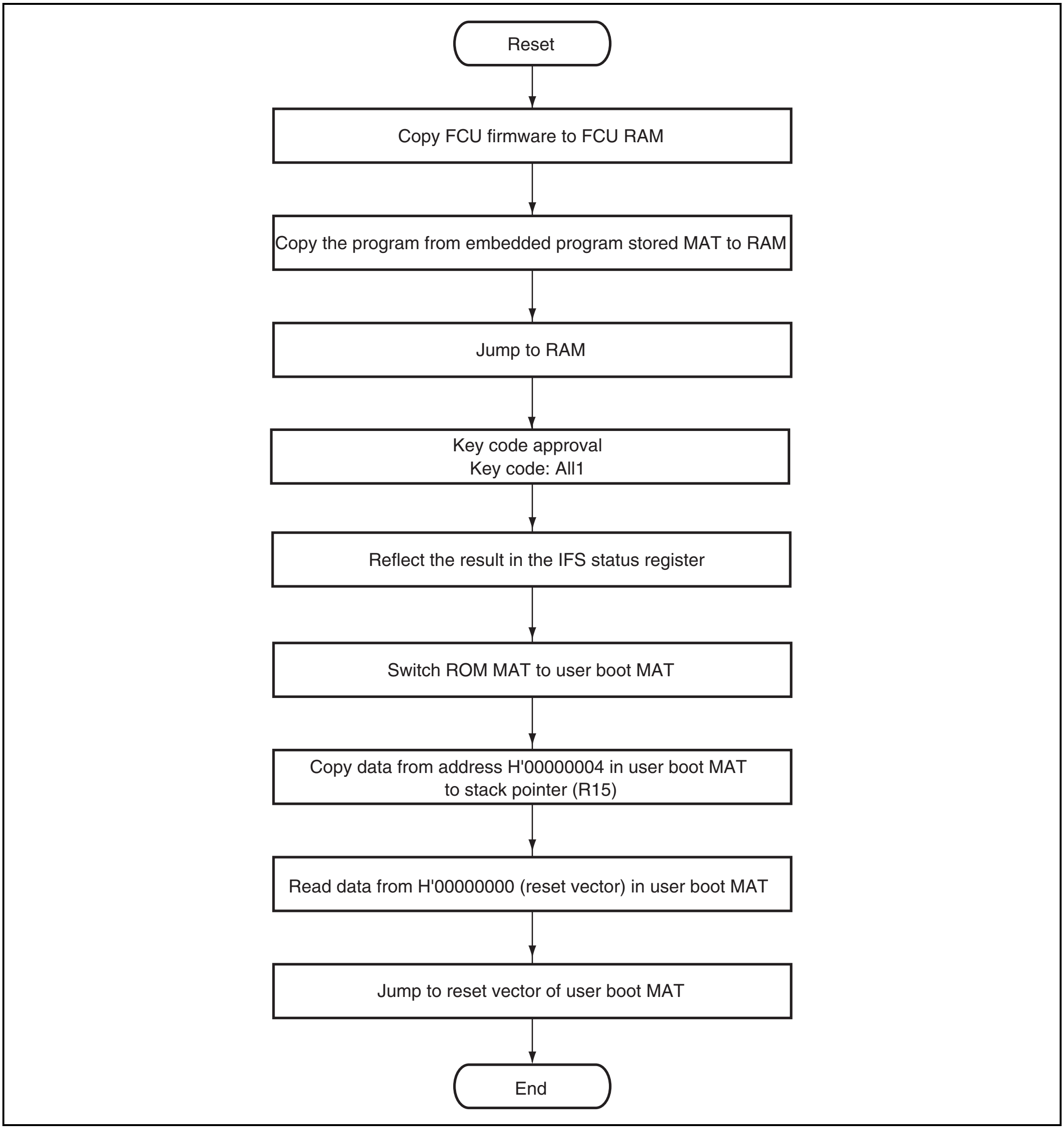

Reset
Copy FCU firmware to FCU RAM
Copy the program from embedded program stored MAT to RAM
Jump to RAM
Key code approval
Key code: All1
Reflect the result in the IFS status register
Switch ROM MAT to user boot MAT
Copy data from address H'00000004 in user boot MAT
to stack pointer (R15)
Read data from H'00000000 (reset vector) in user boot MAT
Jump to reset vector of user boot MAT
End

Figure 27.7 Boot Sequence in User Boot Mode

When this LSI is started in user boot mode, execution starts in the embedded program stored MAT, key code approval (with all bits set to 1) is done during necessary processing such as FCU firmware transfer to the FCU RAM, and then execution jumps to the location indicated by the reset vector of the user boot MAT. At this time, the value corresponding the previously set security setting state is reflected in IFSSR.

Note on User Boot Mode Operation:

In case that this LSI is started in user boot mode, the LSI is started in the previously set security setting state. The security setting state can be checked through the IFS status register.

When programming is performed in the protected-locked state, approve the key code by user software.

- User Program Mode

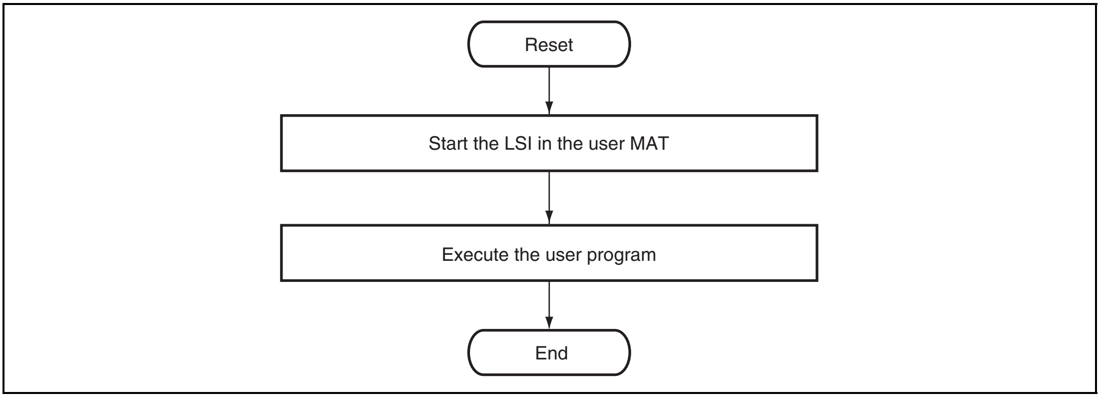


Figure 27.8 Boot Sequence in User Program Mode

When this LSI is started in user program mode, execution starts from the user software in the user MAT (not started in the embedded program stored MAT).

Immediately after start-up, the security is set to the protected-locked state (security level 2). After key code approval with all 64 key code bits set to 1 by user software, the key code setting state matches the IFSST[2:0] bits and the security state of the LSI is determined. When programming is performed in the protected-locked state, approve the key code for the security level set by user software. The security setting state can be checked through the IFS status register.

Note on Operation in User Program Mode:

When the key code approval has failed, the LSI enters the protected-locked state at the previously set security level set.

27.4 Usage Notes

(1) Key Code Management

The key codes set in this LSI should be managed by user applications.

As this IFS function is intended to protect the ROM contents, canceling the security function setting (protected-locked state) and enabling the on-chip ROM programming and erasure is only possible by entering from outside the key code prespecified in this LSI or approving the key code by user software. If the prespecified key code is not known, the security function setting cannot be canceled.

Please note that the user should bear responsibility for protection of the ROM contents regardless of the IFS function setting.

(2) State after Abnormality in Key Code Set or Reset

Table 27.7 shows the security state after an abnormality, such as power failure, occurs during the key code set or reset processing. When the key code set has failed, reset the key code and then set it again.

Table 27.7 State after Abnormality in Key Code Set or Reset

State	Key Code Processing	Key Code Processing Results	State after Return from Abnormal Operation (State after Reset)
Unprotected	Set (security level 1)	The set processing ends abnormally	Unprotected
		The set processing ends correctly	Protected-locked (security level 1)
	Set (security level 2)	The set processing ends abnormally	Unprotected
		The set processing ends correctly	Protected-locked (security level 2)
Protected-unlocked (security level 1)	Reset	The reset processing ends abnormally	Protected-locked (security level 1)
		The reset processing ends correctly	Unprotected
	Set (security level 2)	The set processing ends abnormally	Unprotected Protected-locked (security level 1)
		The set processing ends correctly	Protected-locked (security level 2)
Protected-unlocked (security level 2)	Reset	The reset processing ends abnormally	Protected-locked (security level 1) Protected-locked (security level 2)
		The reset processing ends correctly	Unprotected

(3) Notes on External ROM (CS0 to CS3 Areas) Access

Immediately after this LSI is started in mode 6 (user program mode with the external bus enabled), the security is set to the protected-locked state (security level 2). Because of the security function, correct values cannot be read from the on-chip ROM once an instruction is fetched from the external address space (CS0 to CS3 areas).

To fetch instructions from the external address space in mode 6, enter the protected-unlocked state by key code approval.

Even when the key code has been reset, determine the security state in the LSI through approval with all key code bits set to 1.

(4) Operation on Key Code Approval Failure

Even if key code approval by user software has failed, this LSI keeps the security state corresponding to the previously set key code.

If certain restrictions should be applied (such as transition to the sleep mode) when key code approval has failed, implement the restrictions by user software.

Section 28 EEPROM

This LSI includes 128 Kbytes of flash memory (EEPROM) for storing data. The flash memory has the following features.

28.1 Features

- Flash-memory MATs

  The EEPROM has two types of memory areas (hereafter referred to as memory MATs) in the same address space. These two MATs can be switched by bank switching through the control register. For addresses H'80100080 to H'8011FFFF, the data MAT contents will always be read even when the product information MAT is selected. The product information MAT cannot be programmed or erased.

  Data MAT: 128 Kbytes

  Product information MAT: 128 bytes

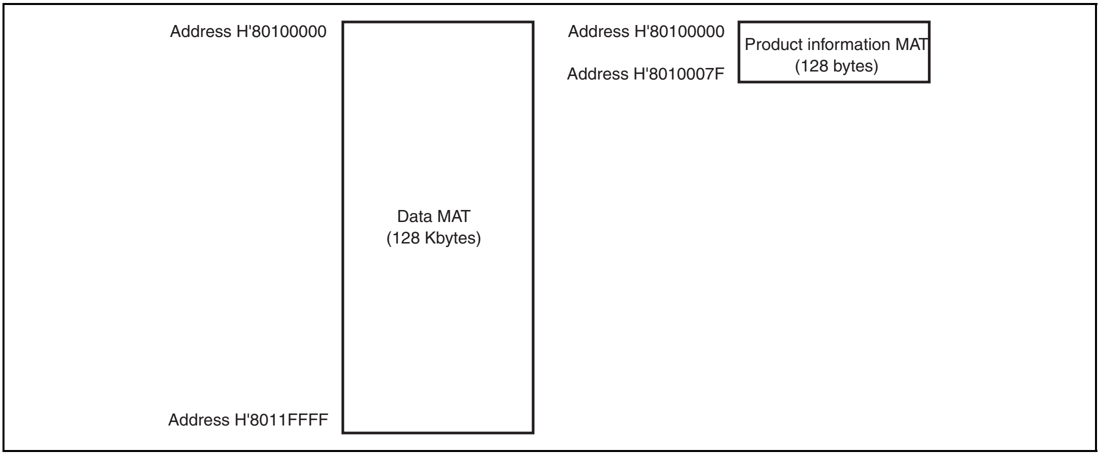


Figure 28.1 Memory MAT Configuration in EEPROM

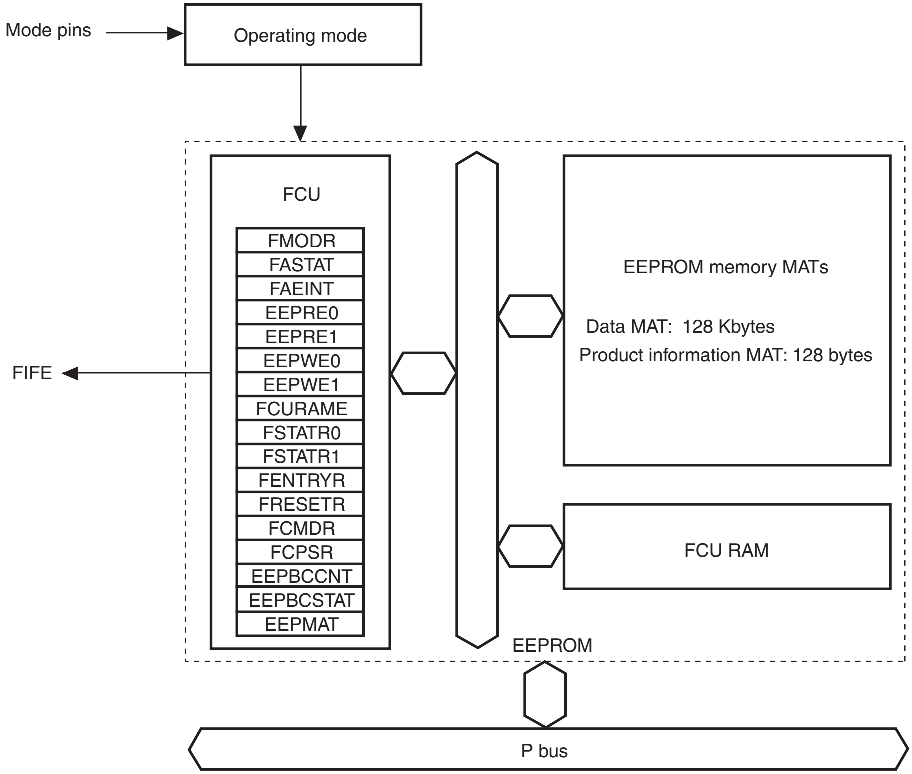
- Reading through the peripheral bus (P bus)

  Both the data MAT and product information MAT can be read through the P bus in three peripheral clock (Pϕ) cycles in words or bytes, or in five Pϕ cycles in longwords, when Pϕ is 16 to 20 MHz. When Pϕ is 32 to 40 MHz, they can be read through the P bus in four Pϕ cycles in words or bytes, or in seven Pϕ cycles in longwords.

- Programming and erasing methods

  The data MAT can be programmed and erased by commands issued through the peripheral bus (P bus) to the ROM/EEPROM-dedicated sequencer (FCU).

  While the FCU is programming or erasing the data MAT, the CPU can execute a program located in the ROM, RAM, or external address space. While the FCU is programming or erasing the ROM or data MAT, data cannot be read form the data MAT. When the FCU suspends programming or erasure, the CPU can read data from the data MAT, and then the FCU can resume programming or erasure of the data MAT. While the FCU suspends erasure, areas other than the erasure-suspended area can be programmed.



[Legend]

FMODR:	Flash mode register
FASTAT:	Flash access status register
FAEINT:	Flash access error interrupt enable register
EEPRE0, EEPRE1:	EEPROM read enable registers 0 and 1
EEPWE0, EEPWE1:	EEPROM program/erase enable registers 0 and 1
FCURAME:	FCU RAM enable register
FSTATR0, FSTATR0:	Flash status registers 0 and 1
FENTRYR:	Flash P/E mode entry register
FRESETR:	Flash reset register
FCMDR:	FCU command register
FCPSR:	FCU processing switch register
EEPBCCNT:	EEPROM blank check control register
EEPBCSTAT:	EEPROM blank check status register
EEPMAT:	EEPROM MAT select register
FIFE:	Flash interface error interrupt

Figure 28.2 Block Diagram of EEPROM

- Programming/erasing unit

  The data MAT is programmed in 8-byte or 128-byte units and erased in block units (8 Kbytes) in user mode, user program mode, and user boot mode. In boot mode, the data MAT is programmed in 256-Kbyte units and erased in block units (8 Kbytes). The product information MAT is read-only memory and cannot be programmed or erased.

  Figure 28.3 shows the block configuration of the data MAT of the LSI. The data MAT is divided into sixteen 8-Kbyte blocks (DB00 to DB15).

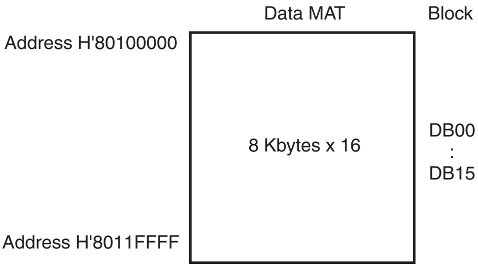


Figure 28.3 Block Configuration of Data MAT

- Blank check function

  If data is read from erased EEPROM by the CPU, undefined values are read. Using blank check command of the FCU allows checking of whether the EEPROM is erased (in a blank state). Either an 8 Kbytes (1 erasure block) or 8 bytes of area can be checked by a single execution of the blank check command.

  Blank check function checks the erase state of the area where erase has ended. The function is disabled when programming/erasing was suspended (e.g., reset input, power-supply interruption).

- Three types of on-board programming modes
  - Boot mode

    The data MAT can be programmed using the SCI. The bit rate for SCI communications between the host and the LSI can be automatically adjusted.
  - User mode/user program mode

    The data MAT can be programmed with a desired interface. The user mode includes the MCU extended mode and MCU single-chip mode (modes 2 and 3) in which the on-chip ROM is enabled.
  - User boot mode

    The data MAT can be programmed with a desired interface. To make a transition to this mode, a reset is needed.

- Protection modes

  This LSI supports two modes to protect memory against programming, erasing, or reading: hardware protection by the levels on the mode pins and software protection by the setting of the FENTRYD bit, EEPRE0 and EEPRE1 registers, or EEPWE0 and EEPWE1 registers. The FENTRYD bit enables or disables data MAT programming or erasure by the FCU. EEPRE0 and EEPRE1 control protection of each data MAT block against reading, and EEPWE0 and EEPWE1 control protection against programming and erasure.

  The LSI also provides a function to suspend programming or erasure when abnormal operation is detected during programming or erasure. In addition, the LSI provides a function to protect the EEPROM against instruction fetch attempted by the CPU.

- Programming and erasing time and count

  Refer to section 34, Electrical Characteristics.

28.2 Input/Output Pins

Table 28.1 shows the input/output pins used for the EEPROM. The combination of MD4 to MD0 pin levels determines the EEPROM programming mode (see section 28.4, Overview of EEPROM-Related Modes). In boot mode, programming and erasing the EEPROM can be performed by the host via the pins PJ6/RxD_A and PJ5/TxD_A (see section 28.5, Boot Mode).

Table 28.1 Pin Configuration

Pin Name	Symbol	I/O	Function
Power-on reset	$\overline{\text{RES}}$	Input	This LSI enters the power-on reset state when this signal goes low.
Mode	MD4 to MD0	Input	These pins specify the operating mode.
Receive data in SCI channel A	PJ6/RxD_A	Input	Receives data through SCI channel A (communications with host)
Transmit data in SCI channel A	PJ5/TxD_A	Output	Transmits data through SCI channel A (communications with host)

28.3 Register Descriptions

Table 28.2 shows the EEPROM-related registers. Some of these registers have ROM-related bits, but this section only describes the EEPROM-related bits. For the registers consisting of bits used by the ROM and EEPROM in common (FCURAME, FSTATR0, FSTATR1, FRESETR, FCMDR, and FCPSR) and the ROM-dedicated bits, refer to section 26.3, Register Descriptions. The EEPROM-related registers are initialized by a power-on reset or a transition to the hardware standby mode.

Table 28.2 Register Configuration

Register Name	Symbol	R/W*[1]	Initial Value	Address	Access Size
Flash mode register	FMODR	R/W	H'00	H'FFFFA802	8
Flash access status register	FASTAT	R/(W)*[2]	H'00	H'FFFFA810	8
Flash access error interrupt enable register	FAEINT	R/W	H'9F	H'FFFFA811	8
EEPROM read enable register 0	EEPRE0	R/(W)*[3]	H'0000	H'FFFFA840	8, 16
EEPROM read enable register 1	EEPRE1	R/(W)*[3]	H'0000	H'FFFFA842	8, 16
EEPROM program/erase enable register 0	EEPWE0	R/(W)*[3]	H'0000	H'FFFFA850	8, 16
EEPROM program/erase enable register 1	EEPWE1	R/(W)*[3]	H'0000	H'FFFFA852	8, 16
FCU RAM enable register	FCURAME	R/(W)*[3]	H'0000	H'FFFFA854	8, 16
Flash status register 0	FSTATR0	R*[5]	H'80	H'FFFFA900	8, 16
Flash status register 1	FSTATR1	R*[5]	H'00	H'FFFFA901	8, 16
Flash P/E mode entry register	FENTRYR	R/(W)*[4]*[5]	H'0000	H'FFFFA902	8, 16
Flash reset register	FRESETR	R/(W)*[3]	H'0000	H'FFFFA906	8, 16
FCU command register	FCMDR	R*[5]	H'FFFF	H'FFFFA90A	8, 16
FCU processing switch register	FCPSR	R/W*[5]	H'0000	H'FFFFA918	8, 16
EEPROM blank check control register	EEPBCCNT	R/W*[5]	H'0000	H'FFFFA91A	8, 16
EEPROM blank check status register	EEPBCSTAT	R*[5]	H'0000	H'FFFFA91E	8, 16
EEPROM MAT select register	EEPMAT	R/(W)*[3]	H'0000	H'FFFFAB00	8, 16

Notes: 1. In on-chip ROM disabled mode, the bits of the EEPROM-related registers are always read as 0 and writing to them is ignored.

2. This register consists of the bits where only 0 can be written to clear the flags and the read-only bits.

3. This register can be written to only when a specified value is written to the upper byte in word access. The data written to the upper byte is not stored in the register.

4. This register can be written to only when a specified value is written to the upper byte in word access; the register is initialized when a value not allowed for the register is written to the upper byte. The data written to the upper byte is not stored in the register.

5. This register can be initialized by a power-on reset, a transition to the hardware standby mode, or by setting the FRESET bit of FRESETR to 1.

28.3.1 Flash Mode Register (FMODR)

FMODR specifies an operating mode for the FCU. In on-chip ROM disabled mode, the FMODR bits are always read as H'00, and writing to them is ignored. FMODR can be initialized by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	FR DMD	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved Write values must always be 0; otherwise normal operation cannot be guaranteed.
4	FRDMD	0	R/W	FCU Read Mode Select Bit Selects the read mode to read the ROM or EEPROM using FCU. This bit specifies the EEPROM lock bit read mode transition or blank check processings in the EEPROM (see section 28.6.1, FCU Command List, and section 28.6.3, FCU Command Usage), whereas this bit must be set to specify the read method for the lock bits in the ROM (see section 26, ROM). 0: Memory area read mode This mode is selected to enter the EEPROM lock bit read mode. Since the EEPROM has no lock bits, reading an EEPROM area results in an undefined value. 1: Register read mode To make the blank check command available for use, register read mode is set.
3 to 0	—	All 0	R	Reserved Write values must always be 0; otherwise normal operation cannot be guaranteed.

28.3.2 Flash Access Status Register (FASTAT)

FASTAT indicates the access error status for the ROM and EEPROM. In on-chip ROM disabled mode, FASTAT is read as H'00 and writing to it is ignored. If any bit in FASTAT is set to 1, the FCU enters command-locked state (see section 28.7.3, Error Protection). To cancel command-locked state, set FASTAT to H'10, and then issue a status-clear command to the FCU. FASTAT is initialized by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	RO MAE	—	—	CM DLK	EE PAE	EEP IFE	EEP RPE	EEP WPE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only 0 can be written to clear the flag after 1 is read.

Bit	Bit Name	Initial Value	R/W	Description
7	ROMAE	0	R/(W)*	ROM Access Error Refer to section 26, ROM.
6, 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	CMDLK	0	R	FCU Command Lock Indicates whether the FCU is in command-locked state (see section 28.7.3, Error Protection). 0: The FCU is not in command-locked state 1: The FCU is in command-locked state [Setting condition] • The FCU detects an error and enters command-locked state. [Clearing condition] • The FCU completes the status-clear command processing.

Bit	Bit Name	Initial Value	R/W	Description
3	EEPAE	0	R/(W)*	EEPROM Access Error Indicates whether an access error has been generated for the EEPROM. If this bit becomes 1, the ILGLERR bit in FSTATR0 is set to 1 and the FCU enters command-locked state. 0: No EEPROM access error has occurred 1: An EEPROM access error has occurred [Setting conditions] • A read access command is issued to the EEPROM area while the FENTRYD bit in FENTRYR is 1 in EEPROM P/E normal mode. • A write access command is issued to the EEPROM area while the FENTRYD bit in FENTRYR is 0. • An access command is issued to the EEPROM area while one of the FENTRY6 to FENTRY0 bits in FENTRYR is 1. [Clearing condition] • 0 is written to this bit after reading EEPAE = 1.
2	EEPIFE	0	R/(W)*	EEPROM Instruction Fetch Error Indicates whether an instruction fetch error has been generated for the EEPROM. 0: No EEPROM instruction fetch error has occurred 1: An EEPROM instruction fetch error has occurred [Setting condition] • An attempt is made to fetch an instruction from the EEPROM. [Clearing condition] • 0 is written to this bit after reading EEPIFE = 1.

Bit	Bit Name	Initial Value	R/W	Description
1	EEPRPE	0	R/(W)*	EEPROM Read Protect Error Indicates whether an error has been generated against the EEPROM read protection provided by the EEPRE0 and EEPRE1 settings. 0: The EEPROM has not been read against the EEPRE0 and EEPRE1 settings 1: An attempt has been made to read data from the EEPROM against the EEPRE0 and EEPRE1 settings [Setting condition] • An attempt is made to read data from the EEPROM area that has been read-protected through the EEPRE0 and EEPRE1 settings. [Clearing condition] • 0 is written to this bit after reading EEPRPE = 1.
0	EEPWPE	0	R/(W)*	EEPROM Program/Erase Protect Error Indicates whether an error has been generated against the EEPROM program/erasure protection provided by the EEPWE0 and EEPWE1 settings. 0: No programming or erasing command has been issued to the EEPROM against the EEPWE0 and EEPWE1 settings 1: A programming or erasing command has been issued to the EEPROM against the EEPWE0 and EEPWE1 settings [Setting condition] • A programming or erasing command is issued to the EEPROM area that has been program/erase-protected through the EEPWE0 and EEPWE1 settings. [Clearing condition] • 0 is written to this bit after reading EEPWPE = 1.

Note: Only 0 can be written to clear the flag after 1 is read.

28.3.3 Flash Access Error Interrupt Enable Register (FAEINT)

FAEINT enables or disables output of flash interface error (FIFE) interrupt requests. In on-chip ROM disabled mode, FAEINT is read as H'00 and writing to it is ignored. FAEINT is initialized by a power-on reset or a transition to the hardware standby mode.

Bit:	7	6	5	4	3	2	1	0
	ROM AEIE	—	—	CMD LKIE	EEP AEIE	EEPI FEIE	EEPR PEIE	EEPW PEIE
Initial value:	1	0	0	1	1	1	1	1
R/W:	R/W	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	ROMAEIE	1	R/W	ROM Access Error Interrupt Enable Refer to section 26, ROM.
6, 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	CMDLKIE	1	R/W	FCU Command Lock Interrupt Enable Enables or disables an FIFE interrupt request when FCU command-locked state is entered and the CMDLK bit in FASTAT becomes 1. 0: Does not generate an FIFE interrupt request when CMDLK = 1 1: Generates an FIFE interrupt request when CMDLK = 1
3	EEPAEIE	1	R/W	EEPROM Access Error Interrupt Enable Enables or disables an FIFE interrupt request when an EEPROM access error occurs and the EEPAE bit in FASTAT becomes 1. 0: Does not generate an FIFE interrupt request when EEPAE = 1 1: Generates an FIFE interrupt request when EEPAE = 1

Bit	Bit Name	Initial Value	R/W	Description
2	EEPIFEIE	1	R/W	EEPROM Instruction Fetch Error Interrupt enable Enables or disables an FIFE interrupt request when an EEPROM instruction fetch error occurs and the EEPIFE bit in FASTAT becomes 1. 0: Does not generate an FIFE interrupt request when EEPIFE = 1 1: Generates an FIFE interrupt request when EEPIFE = 1
1	EEPRPEIE	1	R/W	EEPROM Read Protect Error Interrupt Enable Enables or disables an FIFE interrupt request when an EEPROM read protect error occurs and the EEPRPE bit in FASTAT becomes 1. 0: Does not generate an FIFE interrupt request when EEPRPE = 1 1: Generates an FIFE interrupt request when EEPRPE = 1
0	EEPWPEIE	1	R/W	EEPROM Program/Erase Protect Error Interrupt Enable Enables or disables an FIFE interrupt request when an EEPROM program/erase protect error occurs and the EEPWPE bit in FASTAT becomes 1. 0: Does not generate an FIFE interrupt request when EEPWPE = 1 1: Generates an FIFE interrupt request when EEPWPE = 1

28.3.4 EEPROM Read Enable Register 0 (EEPRE0)

EEPRE0 enables or disables read access to blocks DB00 to DB07 (see figure 28.3) in the data MAT. In on-chip ROM disabled mode, EEPRE0 is read as H'0000 and writing to it is ignored. EEPRE0 is initialized by a power-on reset or a transition to the hardware standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KEY								DBRE07	DBRE06	DBRE05	DBRE04	DBRE03	DBRE02	DBRE01	DBRE00
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Written data is not stored in these bits.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	KEY	H'00	R/(W)*	Key Code These bits enable or disable DBRE07 to DBRE00 bit modification. The data written to these bits are not stored.
7	DBRE07	0	R/W	DB07 to DB00 Block Read Enable Enables or disables read access to blocks DB07 to DB00 in the data MAT. The DBREi bit (i = 07 to 00) controls read access to block DBi. Writing to these bits is enabled only when this register is accessed in word size and H'2D is written to the KEY bits. 0: Disables read access 1: Enables read access
6	DBRE06	0	R/W	
5	DBRE05	0	R/W	
4	DBRE04	0	R/W	
3	DBRE03	0	R/W	
2	DBRE02	0	R/W	
1	DBRE01	0	R/W	
0	DBRE00	0	R/W	

28.3.5 EEPROM Read Enable Register 1 (EEPRE1)

EEPRE1 enables or disables read access to blocks DB08 to DB15 (see figure 28.3) in the data MAT. In on-chip ROM disabled mode, EEPRE1 is read as H'0000 and writing to it is ignored. EEPRE1 is initialized by a power-on reset or a transition to the hardware standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KEY								DBRE15	DBRE14	DBRE13	DBRE12	DBRE11	DBRE10	DBRE09	DBRE08
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Written data is not stored in these bits.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	KEY	H'00	R/(W)*	Key Code These bits enable or disable DBRE15 to DBRE08 bit modification. The data written to these bits are not stored.
7	DBRE15	0	R/W	DB15 to DB08 Block Read Enable Enables or disables read access to blocks DB15 to DB08 in the data MAT. The DBREi bit (i = 15 to 08) controls read access to block DBi. Writing to these bits is enabled only when this register is accessed in word size and H'D2 is written to the KEY bits. 0: Disables read access 1: Enables read access
6	DBRE14	0	R/W	
5	DBRE13	0	R/W	
4	DBRE12	0	R/W	
3	DBRE11	0	R/W	
2	DBRE10	0	R/W	
1	DBRE09	0	R/W	
0	DBRE08	0	R/W	

28.3.6 EEPROM Program/Erase Enable Register 0 (EEPWE0)

EEPWE0 enables or disables programming and erasure of blocks DB00 to DB07 (see figure 28.3) in the data MAT. In on-chip ROM disabled mode, EEPWE0 is read as H'0000 and writing to it is ignored. EEPWE0 is initialized by a power-on reset or a transition to the hardware standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KEY								DBWE07	DBWE06	DBWE05	DBWE04	DBWE03	DBWE02	DBWE01	DBWE00
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Written data is not stored in these bits.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	KEY	H'00	R/(W)*	Key Code These bits enable or disable DBWE07 to DBWE00 bit modification. The data written to these bits are not stored.
7	DBWE07	0	R/W	DB07 to DB00 Block Program/Erase Enable Enables or disables programming and erasure of blocks DB07 to DB00 in the data MAT. The DBWEi bit (i = 07 to 00) controls programming and erasure of block DBi. Writing to these bits is enabled only when this register is accessed in word size and H'1E is written to the KEY bits. 0: Disables programming and erasure 1: Enables programming and erasure
6	DBWE06	0	R/W	
5	DBWE05	0	R/W	
4	DBWE04	0	R/W	
3	DBWE03	0	R/W	
2	DBWE02	0	R/W	
1	DBWE01	0	R/W	
0	DBWE00	0	R/W	

28.3.7 EEPROM Program/Erase Enable Register 1 (EEPWE1)

EEPWE1 enables or disables programming and erasure of blocks DB08 to DB15 (see figure 28.3) in the data MAT. In on-chip ROM disabled mode, EEPWE1 is read as H'0000 and writing to it is ignored. EEPWE1 is initialized by a power-on reset or a transition to the hardware standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KEY								DBWE15	DBWE14	DBWE13	DBWE12	DBWE11	DBWE10	DBWE09	DBWE08
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Written data is not stored in these bits.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	KEY	H'00	R/(W)*	Key Code These bits enable or disable DBWE15 to DBWE08 bit modification. The data written to these bits are not stored.
7	DBWE15	0	R/W	DB15 to DB08 Block Program/Erase Enable Enables or disables programming and erasure of blocks DB15 to DB08 in the data MAT. The DBWEi bit (i = 15 to 08) controls read access to block DBi. Writing to these bits is enabled only when this register is accessed in word size and H'E1 is written to the KEY bits. 0: Disables programming and erasure 1: Enables programming and erasure
6	DBWE14	0	R/W	
5	DBWE13	0	R/W	
4	DBWE12	0	R/W	
3	DBWE11	0	R/W	
2	DBWE10	0	R/W	
1	DBWE09	0	R/W	
0	DBWE08	0	R/W	

28.3.8 Flash P/E Mode Entry Register (FENTRYR)

FENTRYR specifies the P/E mode for the ROM or EEPROM. To specify the P/E mode for the ROM or EEPROM so that the FCU can accept commands, set FENTRYD and FENTRY6–FENTRY0 to 1. Note that if this register is set to other than H'0001, H'0002, H'0004, H'0008, H'0010, H'0020, or H'0080, the ILGLERR bit in the FSTATR0 register will be set and the FCU will enter command-locked state. In on-chip ROM disabled mode, FENTRYR is read as H'0000 and writing to it is ignored. FENTRYR is initialized by a power-on reset, a transition to the hardware standby mode, or setting the FRESET bit of FRESETR to 1.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FEKEY								FEN TRYD	FEN TRY6	FEN TRY5	FEN TRY4	FEN TRY3	FEN TRY2	FEN TRY1	FEN TRY0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Written data is not stored in these bits.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	FEKEY	H'00	R/(W)*	Key Code These bits enable or disable the FENTRYD and FENTRY6 to FENTRY0 bit modification. The data written to these bits are not stored.

Bit	Bit Name	Initial Value	R/W	Description
7	FENTRYD	0	R/W	EEPROM P/E Mode Entry This bit specifies the P/E mode for the EEPROM. 0: The EEPROM is in read mode 1: The EEPROM is in P/E mode [Write enabling conditions] When the following conditions are all satisfied: • The LSI is in on-chip ROM enabled mode. • The FRDY bit in FSTATR0 is 1. • H'AA is written to FEKEY in word access. [Setting condition] • 1 is written to FENTRYD while the write enabling conditions are satisfied and FENTRYR is H'0000. [Clearing conditions] • This register is written to in byte access. • A value other than H'AA is written to FEKEY in word access. • 0 is written to FENTRYD while the write enabling conditions are satisfied. • FENTRYR is written to while FENTRYR is not H'0000 and the write enabling conditions are satisfied.
6 to 0	FENTRY6 to FENTRY0	All 0	R/W	ROM P/E Mode Entry 6 to 0 Refer to section 26, ROM.

28.3.9 EEPROM Blank Check Control Register (EEPBCCNT)

EEPBCCNT specifies the addresses and sizes of the target areas to be checked by the blank check command. In on-chip ROM disabled mode, EEPBCCNT is read as H'0000, and writing to it is ignored. EEPBCCNT is initialized by a power-on reset, a transition to the hardware standby mode, or by setting the FRESET bit of FRESETR to 1.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	BCADR										-	-	BC SIZE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	All 0	R	Reserved The write value must always be 0; otherwise operation is not guaranteed.
12 to 3	BCADR	All 0	R/W	Blank Check Address Setting Bit Use these bits to specify the address of the target area when the size of the target area to be checked by the blank check command is 8 bytes (the BCSIZE bit is set to 0). When the BCSIZE bit is set to 0, the start address of the target area is the value obtained by summing the EEPBCCNT value (the value obtained by shifting the set BCADR value by 3 bits) and the start address of an erased block specified when a blank check command is issued.
2, 1	—	All 0	R	Reserved The write value must always be 0; otherwise operation is not guaranteed.
0	BCSIZE	0	R/W	Blank Check Size Setting Bit This bit selects the size of the target area to be checked by the blank check command. 0: Selects 8 bytes as the size of a blank check target area. 1: Selects 8 Kbytes as the size of a blank check target area.

28.3.10 EEPROM Blank Check Status Register (EEPBCSTAT)

EEPBCSTAT stores check results by executing the blank check command. In on-chip ROM disabled mode, EEPBCSTAT is read as H'0000, and writing to it is ignored. EEPBCSTAT is initialized by a power-on reset, a transition to the hardware standby mode, or by setting the FRESET bit of FRESETR to 1.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	BCST
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 1	—	All 0	R	Reserved The write value must be 0; otherwise operation is not guaranteed.
0	BCST	0	R	Blank Check Status Bit Indicates the result of a blank check. 0: The target area is erased (blank). 1: The target area is filled with 0s and/or 1s.

28.3.11 EEPROM MAT Select Register (EEPMAT)

EEPMAT switches memory MATs in the EEPROM. In on-chip ROM disabled mode, EEPMAT is read as H'0000 and writing to it is ignored. EEPMAT is initialized by a power-on reset or a transition to the hardware standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KEY								—	—	—	—	—	—	—	EEP SEL
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R	R	R	R	R	R	R	R/W

Note: *1 Written data is not stored in these bits.

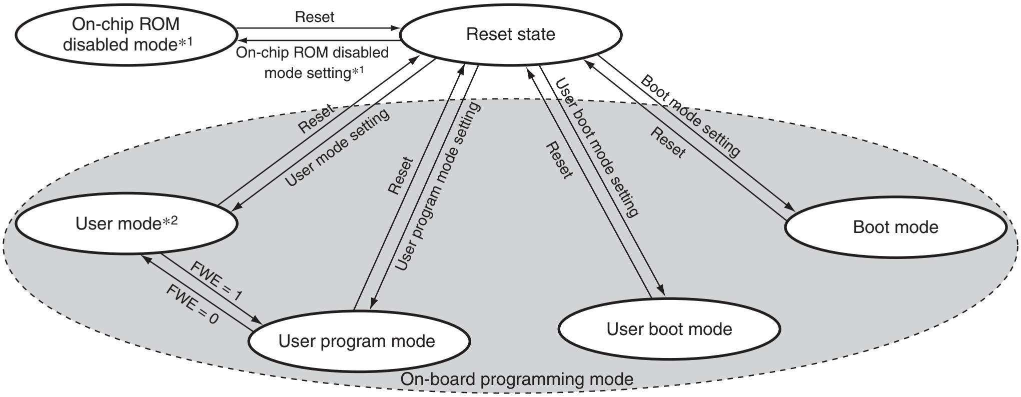
Bit	Bit Name	Initial Value	R/W	Description
15 to 8	KEY	H'00	R/(W)*1	Key Code These bits enable or disable EEPSEL bit modification. The data written to these bits are not stored.
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	EEPSEL	0	R/W	EEPROM MAT Select Selects a memory MAT in the EEPROM. Writing to this bit is enabled only when this register is accessed in word size and H'B3 is written to the KEY bits. 0: Selects the data MAT 1: Selects the product information MAT*2

Notes: 1. Written data is not stored in these bits.

2. The product information MAT is read-only memory and cannot be programmed or erased.

28.4 Overview of EEPROM-Related Modes

Figure 28.4 shows the EEPROM-related mode transition in this LSI. For the relationship between the LSI operating modes and the MD4 to MD0 and FWE pin settings, refer to section 3, Operating Modes.



Notes: 1. Modes 0 and 1 (MCU extended modes) of this LSI are on-chip ROM disabled modes.
2. Mode 2 (MCU extended mode) and mode 3 (MCU single-chip mode) of this LSI are user modes.

Figure 28.4 EEPROM-Related Mode Transition

- The EEPROM cannot be read, programmed, or erased in on-chip ROM disabled mode.
- The data MAT can be read, programmed, and erased on the board in user mode, user program mode, user boot mode, and boot mode.
- In user mode, the ROM cannot be programmed or erased but the EEPROM can be programmed and erased. While the EEPROM is being programmed or erased, the ROM can be read. Therefore, the user can program the EEPROM while executing an application program in the ROM protected against programming and erasure.

Table 28.3 compares programming- and erasure-related items for the boot mode, user mode, user program mode, and user boot mode.

Table 28.3 Comparison of Programming Modes

Item	Boot Mode	User Mode	User Program Mode	User Boot Mode
Programming/erasure enabled MAT	Data MAT	Data MAT	Data MAT	Data MAT
Programming/erasure control	Host	FCU	FCU	FCU
Programming data transfer	From host via SCI	From any device via RAM	From any device via RAM	From any device via RAM
Reset-start MAT	Embedded program stored MAT	User MAT	User MAT	User boot MAT*

Note: * After the LSI is started in the embedded program stored MAT and the embedded program built in the product is executed, execution starts from the location indicated by the reset vector of the user boot MAT.

- In user boot mode, a boot operation with a desired interface can be implemented through mode pin settings different from those in user mode or user program mode.
- Both in boot mode and user boot mode, the embedded program built in the product uses H'FFF88000 to H'FFF8FFFF in the internal RAM. Therefore, once the RAM is disabled via the RAM Enable Control register (RAMEN) and a reset is issued, the data stored in the corresponding area in the RAM prior to the reset is no longer stored in the RAM after booting is initiated in boot mode or user boot mode (see section 30, RAM).

28.5 Boot Mode

To program or erase the data MAT in boot mode, send control commands and programming data from the host. For the system configuration and settings in boot mode, refer to section 26, ROM. This section describes only the commands dedicated for the EEPROM.

28.5.1 Inquiry/Selection Host Commands

Table 28.4 shows the inquiry/selection host commands dedicated to the EEPROM. The data MAT inquiry and data MAT information inquiry commands are used in the step for inquiry regarding the MAT programming information shown in figure 26.9 in section 26.5.4, Inquiry/Selection Host Command Wait State.

Table 28.4 Inquiry/Selection Host Commands (for EEPROM only)

Host Command Name	Function
Data MAT inquiry	Inquires regarding the availability of user MAT
Data MAT information inquiry	Inquires regarding the number of data MATs and the start and end addresses

Each host command is described in detail below. The "command" in the description indicates a command sent from the host to this LSI and the "response" indicates a response sent from this LSI to the host. The "checksum" is byte-size data calculated so that the sum of all bytes to be sent by this LSI becomes H'00.

(1) Data MAT Inquiry

In response to a data MAT inquiry command sent from the host, this LSI returns the information concerning the availability of data MATs.

Command	H'2A

Response	H'3A	Size	Availability	SUM

[Legend]

Size (1 byte): Total number of characters in the availability field (fixed at 1)

Availability (1 byte): Availability of data MATs (fixed at H'01)
H'00: No data MAT is available
H'01: Data MAT is available

SUM (1 byte): Checksum

(2) Data MAT Information Inquiry

In response to a data MAT information inquiry command sent from the host, this LSI returns the number of data MATs and their addresses.

Command

H'2B	

Response

H'3B	Size	MAT count
MAT start address		
MAT end address		
MAT start address		
MAT end address		
:		
MAT start address		
MAT end address		
SUM		

[Legend]

Size (1 byte): Total number of bytes in the MAT count, MAT start address, and MAT end address fields

MAT count (1 byte): Number of data MATs (consecutive areas are counted as one MAT)

MAT start address (4 bytes): Start address of a data MAT

MAT end address (4 bytes): End address of a data MAT

SUM (1 byte): Checksum

The information concerning the block configuration in the data MAT is included in the response to the erasure block information inquiry command (refer to section 26.5.4, Inquiry/Selection Host Command Wait State).

28.5.2 Programming/Erasing Host Commands

Table 28.5 shows the programming/erasing host commands dedicated to the EEPROM. EEPROM-dedicated host commands are provided only for checksum and blank check; the programming, erasing, and reading commands are used in common for the ROM and EEPROM.

To program the data MAT, issue from the host a user MAT programming selection command and then a 256-byte programming command specifying a data MAT address as the programming address. To erase the data MAT, issue an erasure selection command and then a block erasure command specifying an erasure block in the data MAT. The information concerning the erasure block configuration in the data MAT is included in the response to the erasure block information inquiry command. To read data from the data MAT, select the user MAT through a memory read command specifying a data MAT address as the read address.

For the user MAT programming selection, user boot MAT programming selection, 256-byte programming, erasure selection, block erasure selection, and memory read commands, refer to section 26.5.5, Programming/Erasing Host Command Wait State. For the erasure block information inquiry command, refer to section 26.5.4, Inquiry/Selection Host Command Wait State.

Table 28.5 Programming/Erasure Host Commands (for EEPROM)

Host Command Name	Function
Data MAT checksum	Performs checksum verification for the data MAT
Data MAT blank check	Checks whether the data MAT is blank

Each host command is described in detail below. The "command" in the description indicates a command sent from the host to this LSI and the "response" indicates a response sent from this LSI to the host. The "checksum" is byte-size data calculated so that the sum of all bytes to be sent by this LSI becomes H'00.

(1) Data MAT Checksum

In response to a data MAT checksum command sent from the host, this LSI sums the data MAT data in byte units and returns the result (checksum).

Command	H'61

Response	H'71	Size	MAT checksum	SUM

[Legend]

Size (1 byte): Number of bytes in the MAT checksum field (fixed at 4)

MAT checksum (4 bytes): Checksum of the data MAT data

SUM (1 byte): Checksum (for the response data)

(2) Data MAT Blank Check

In response to a data MAT blank check command sent from the host, this LSI checks whether the data MAT is completely erased. When the data MAT is completely erased, this LSI returns a response (H'06). If the user MAT has an unerased area, this LSI returns an error response (sends H'E2 and H'52 in that order).

Command	H'62

Response	H'06

Error response	H'E2	H'52

28.6 User Mode, User Program Mode, and User Boot Mode

28.6.1 FCU Command List

To program or erase the data MAT in user mode, user program mode, or user boot mode, issue FCU commands to the FCU. Table 28.6 is a list of FCU commands for EEPROM programming and erasure.

Table 28.6 FCU Command List (EEPROM-Related Commands)

Command	Function
Normal mode transition	Moves to the normal mode (see section 28.6.2, Conditions for FCU Command Acceptance).
Status read mode transition	Moves to the status read mode (see section 28.6.2, Conditions for FCU Command Acceptance).
Lock bit read mode transition (lock bit read 1)	Moves to the lock bit read mode (see section 28.6.2, Conditions for FCU Command Acceptance).
Program	Programs EEPROM (in 8-byte or 128-byte units).
Block erase	Erases EEPROM (in block units).
P/E suspend	Suspends programming or erasure.
P/E resume	Resumes programming or erasure.
Status register clear	Clears the IRGERR, ERSERR, and PRGERR bits in FSTATR0 and cancels the command-locked state.
Blank check	Checks if a specified area is erased (blank).

FCU commands other than the program command and blank check command are also used for ROM programming and erasure. When the blank check command is issued to the ROM, the lock bits in the ROM are read out.

To issue a command to the FCU, access the EEPROM area through the P bus. Table 28.7 shows the FCU command formats for the program command and blank check command. For the other command formats, refer to section 26.6.1, FCU Command List. When a P-bus access, as shown in table 28.7, is made under specified conditions, the FCU performs processing specified by a selected command. For the conditions for the FCU command acceptance, refer to section 28.6.2, Conditions for FCU Command Acceptance. For details of command usage, refer to section 28.6.3, FCU Command Usage.

When the FRDMD bit is set to 0 (memory area read mode), if the data in the first cycle of an FCU command is determined as H'71, the FCU accepts the lock bit read mode transition command. Since the EEPROM has no lock bits, making P-bus access after a transition to the lock bit read mode results in undefined read data. The FCU detects no access violation error when the undefined data is read. When the FRDMD bit is set to 1 (register read mode), if the data in the first cycle of an FCU command is determined as H'71, the FCU enters a waiting state to wait for the command in the second cycle (H'D0) of the blank check command. At this stage, if H'D0 is written into an EEPROM area by a P-bus write access, the FCU detects it and starts performing the blank check processes specified by the set values in the EEPBCCNT register, and once the check completes the FCU writes check results into the EEPBCSTAT register.

There are two suspending modes to be initiated by the P/E suspend command; the suspension-priority mode and erasure-priority mode. For details of each mode, refer to section 26.6.4, Suspending Operation.

Table 28.7 FCU Command Formats (for EEPROM only)

	Number of Bus	First Cycle		Second Cycle		Third Cycle		Fourth Cycle to Cycle N + 2		Cycle N + 3	
Command	Cycles	Address	Data	Address	Data	Address	Data	Address	Data	Address	Data
Program (8-byte programming: N = 4)	7	EA	H'E8	EA	H'04	WA	WD1	EA	WDn	EA	H'D0
Program (128-byte programming: N = 64)	67	EA	H'E8	EA	H'40	WA	WD1	EA	WDn	EA	H'D0
Blank check	2	EA	H'71	BA	H'D0	—	—	—	—	—	—

[Legend]

EA: EEPROM area address
An arbitrary address within the range of H'8010000 to H'8011FFFF

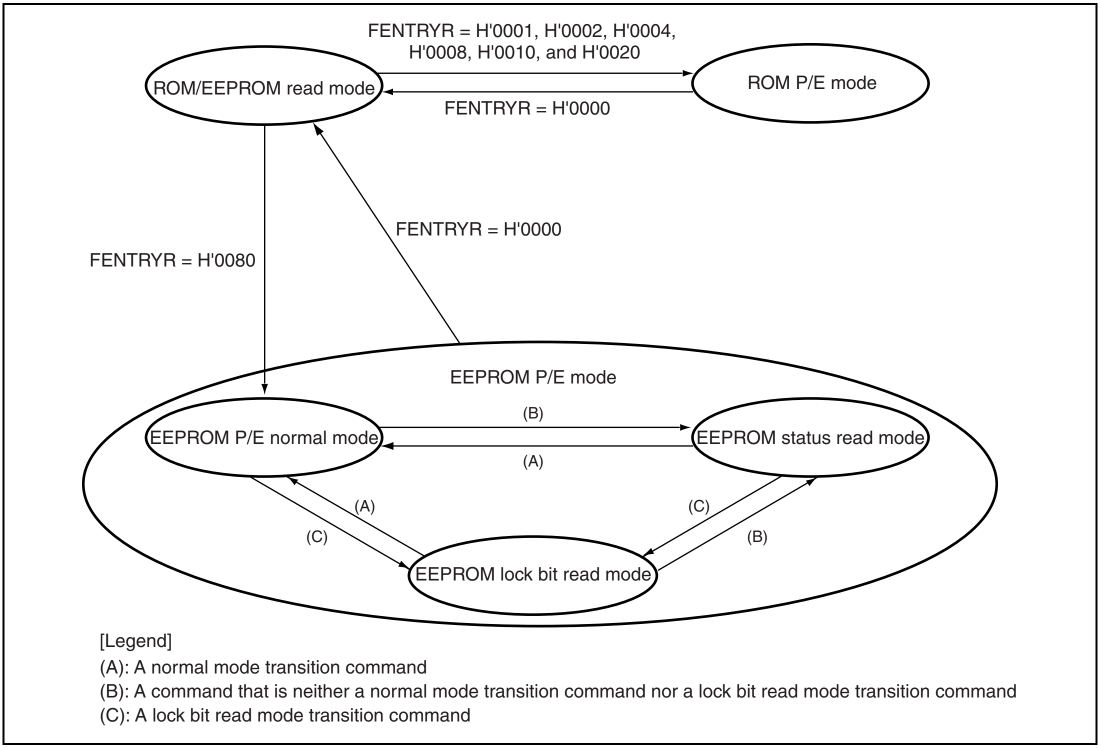
WA: The start address of write data

BA: The address of an EEPROM erasure block
(An arbitrary address in the erase target block)

WDn: n-th word of programming data (n = 1 to N)

28.6.2 Conditions for FCU Command Acceptance

The FCU determines whether to accept a command depending on the FCU mode or status. Figure 28.5 is an FCU mode transition diagram.



[Legend]
(A): A normal mode transition command
(B): A command that is neither a normal mode transition command nor a lock bit read mode transition command
(C): A lock bit read mode transition command

Figure 28.5 FCU Mode Transition Diagram (EEPROM-Related Modes)

(1) ROM P/E Mode

The FCU can accept ROM programming and erasing commands in this mode. The EEPROM cannot be read. The FCU enters this mode when the FENTRYD bit is set to 0 and any one of the bits, FENTRY6 to FENTRY0, is set to 1 in FENTRYR. For details of this mode, refer to section 26.6.2, Conditions for FCU Command Acceptance.

(2) ROM/EEPROM Read Mode

The EEPROM can be read through the peripheral bus, and the ROM can be read through the ROM cache at a high speed. The FCU does not accept commands. The FCU enters this mode when the FENTRY6 to FENTRY0 bits are set to 0000000 and the FENTRYD bit to 0 in FENTRYR.

(3) EEPROM P/E Mode

- EEPROM P/E normal mode

  The FCU enters this mode when the FENTRYD bit is set to 1 and the FENTRY6 to FENTRY0 bits are set to 0000000 in ROM/EEPROM read mode or ROM P/E mode, or when a normal mode transition command is accepted in EEPROM P/E mode. Table 28.8 shows the commands that can be accepted in this mode. If the EEPROM area is read through the P bus, an EEPROM access error occurs and the FCU enters the command-locked state.
- EEPROM status read mode

  The FCU enters this mode when the FCU accepts a command that is neither the normal mode transition command nor the lock bit read mode transition command in EEPROM P/E mode. The EEPROM status read mode includes the state in which the FRDY bit in FSTATR0 is 0 and the command-locked state after an error has occurred. Table 28.8 shows the commands that can be accepted in this mode. If the EEPROM area is read through the P bus, the FSTATR0 value is read.
- EEPROM lock bit read mode

  The FCU enters this mode when the FCU accepts a lock bit read mode transition command in EEPROM P/E mode. Table 28.8 shows the commands that can be accepted in this mode. Since the EEPROM has no lock bits, reading an EEPROM area via the P-bus results in an undefined value. However, no access violation occurs in this case.

Table 28.8 shows the correlation between each FCU mode and its register /state and its acceptable commands. When an unacceptable command is issued, the FCU enters the command-locked state (see section 28.7.3, Error Protection).

To make sure that the FCU accepts a command, enter the mode in which the FCU can accept the target command, check the FRDY, ILGLERR, ERSERR, and PRGERR bit values in FSTATR0, and the FCUERR, FRDTCT, and FRCRCT bit values in FSTATR1, and then issue the target FCU command. The CMDLK bit in FASTAT holds a value obtained by logical ORing the ILGLERR, ERSERR, and PRGERR bit values in FSTATR0 and the FCUERR, FRDTCT, and FRCRCT bit values in the FSTATR1. Therefore the FCU's error occurrence state can be checked by reading the CMDLK bit. In table 28.8, the CMDLK bit is used as the bit to indicate the error occurrence state. The FRDY bit of FSTATR0 is 0 during the programming/erasure, programming/erasure suspension, and blank check processes. While the FRDY bit is 0, the P/E suspend command can be accepted only when the SUSRDY bit in FSTATR0 is 1.

Table 28.8 includes 0 and 1 in single cells of the ERSSPD, PRGSPD, and FRDY bit rows for the sake of simplification. The ERSSPD bits 1 and 0 indicate the erasure suspension and programming suspension processes, respectively. The PRGSPD bits 1 and 0 indicate the programming suspension and erasure suspension processes, respectively. The FRDY bit value can be either 1 or 0, which is a value held by the bit prior to a transition to the command lock state.

Table 28.8 FCU Modes/States and Acceptable Commands

	P/E Normal Mode			Status Read Mode									Lock Bit Read Mode		
Item	Programming-Suspended	Erasure-Suspended	Other State	Programming/Erasure Processing	Programming Processing during Erasure-Suspended	Programming/Erasure Suspension Processing	Blank Check Processing	Programming-Suspended	Erasure-Suspended	Command-Locked (FRDY = 0)	Command-Locked (FRDY = 1)	Other State	Programming-Suspended	Erasure-Suspended	Other State
FRDY bit in FSTATR0	1	1	1	0	0	0	0	1	1	0	1	1	1	1	1
SUSRDY bit in FSTATR0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
ERSSPD bit in FSTATR0	0	1	0	0	1	0/1	0/1	0	1	0/1	0/1	0	0	1	0
PRGSPD bit in FSTATR0	1	0	0	0	0	0/1	0/1	1	0	0/1	0/1	0	1	0	0
CMDLK bit in FASTAT	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
Normal mode transition	A	A	A	×	×	×	×	A	A	×	×	A	A	A	A
Status read mode transition	A	A	A	×	×	×	×	A	A	×	×	A	A	A	A
Lock bit read mode transition (lock bit read 1)	A	A	A	×	×	×	×	A	A	×	×	A	A	A	A
Program	×	*	A	×	×	×	×	×	*	×	×	A	×	*	A
Block erase	×	×	A	×	×	×	×	×	×	×	×	A	×	×	A
P/E suspend	×	×	×	A	×	×	×	×	×	×	×	×	×	×	×
P/E resume	A	A	×	×	×	×	×	A	A	×	×	×	A	A	×
Status register clear	A	A	A	×	×	×	×	A	A	×	A	A	A	A	A
Blank check	A	A	A	×	×	×	×	A	A	×	×	A	A	A	A

[Legend]

A: Acceptable

*: Only programming is acceptable for the areas other than the erasure-suspended block

×: Not acceptable

28.6.3 FCU Command Usage

This section shows how to program and erase the EEPROM using the program command and block erase command, respectively, and how to check the erasure status of the EEPROM using the blank check command. For the firmware transfer to the FCU RAM and the other FCU command usage, refer to section 26.6.3, FCU Command Usage.

If the FCU enters the command lock state in the middle of its handling of commands by setting the FCUERR, FRDTCT, or FRCRCT bit in FSTATR1 to 1, the FRDY bit in FSTATR0 retains 0. Since the FCU halts its operation in the command lock state, the FRDY bit is not set to 1 from 0.

If the FRDY bit retains 0 for longer than the programming/erasure time or suspend delay time (see section 34, Electrical Characteristics), an abnormal operation may have occurred. In such case, initialize the FCU by issuing an FCU reset.

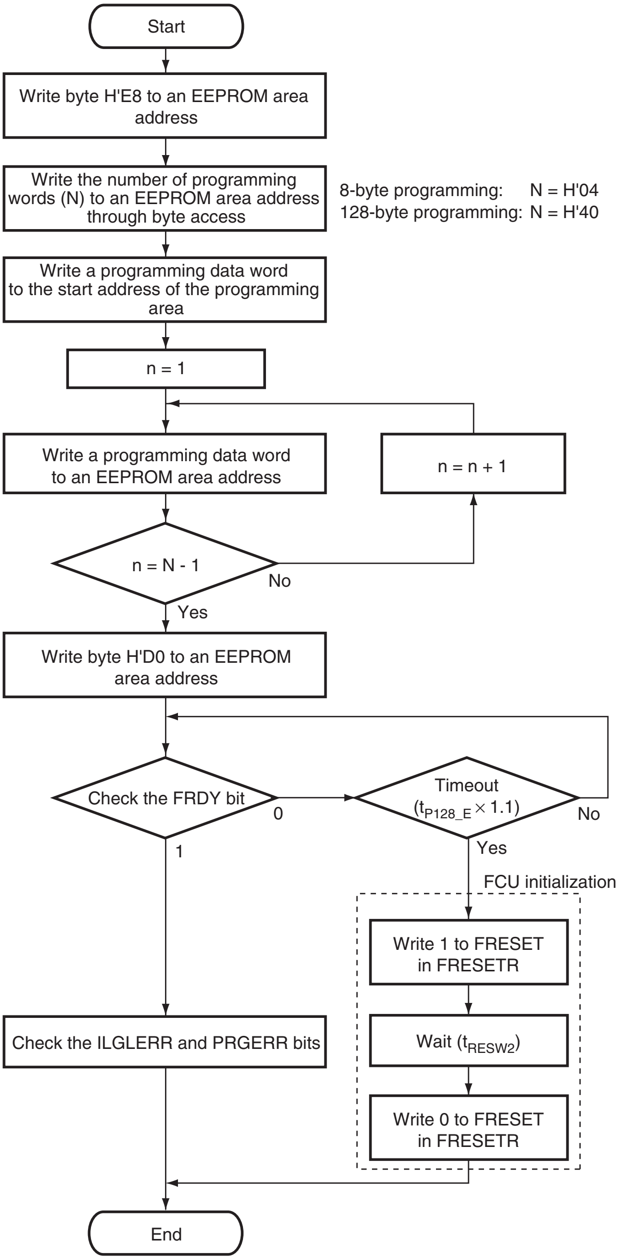
If the FRDY bit is set to 1 upon the termination of an FCU command operation, both the FCUERR, FRDTCT, and FRCRCT bits are cleared to 0. On the other hand, it can be checked via the ILGLERR, ERSERR, or PRGERR bit whether or not an error has occurred after a command operation terminates.

(1) Programming

To program the EEPROM, use the program command. Write byte H'E8 to an EEPROM area address in the first cycle of the program command and the number of words (N)* to be programmed through byte access in the second cycle. Access the P bus in words from the third cycle to cycle N + 2 of the command. In the third cycle, write the programming data to the start address of the target programming area. Here, the start address must be an 8-byte boundary address for 8-byte programming or a 128-byte boundary address for 128-byte programming. After writing words to EEPROM area addresses N times, write byte H'D0 to an EEPROM area address in cycle N + 3; the FCU then starts EEPROM programming. Read the FRDY bit in FSTATR0 to confirm that EEPROM programming is completed.

If the area accessed in the third cycle to cycle N + 2 includes addresses that do not need to be programmed, write H'FFFF as the programming data for those addresses. To ignore the programming and erasure protection provided by the EEPWE0 and EEPWE1 settings, set the program/erase enable bit for the target block to 1 before starting programming. To ignore the protection provided by the lock bit during programming, set the FPROTCN bit in FPROTR to 1 before starting programming. Figure 28.6 shows the procedure for EEPROM programming

Note: * N = H'04 for 8-byte programming or N = H'40 for 128-byte programming.


Start
Write byte H'E8 to an EEPROM area address
Write the number of programming words (N) to an EEPROM area address through byte access
8-byte programming: N = H'04
128-byte programming: N = H'40
Write a programming data word to the start address of the programming area
n = 1
Write a programming data word to an EEPROM area address
n = n + 1
n = N - 1
No
Yes
Write byte H'D0 to an EEPROM area address
Check the FRDY bit
0
1
Timeout (t_{P128_E} × 1.1)
No
Yes
FCU initialization
Write 1 to FRESET in FRESETR
Wait (t_{RESW2})
Write 0 to FRESET in FRESETR
Check the ILGLERR and PRGERR bits
End

Notes: t_{P128_E}: Time required for programming 128-byte data (see section 34, Electrical Characteristics).
t_{RESW2}: Reset pulse width during programming and erasure (see section 34, Electrical Characteristics).

Figure 28.6 Procedure for EEPROM Programming

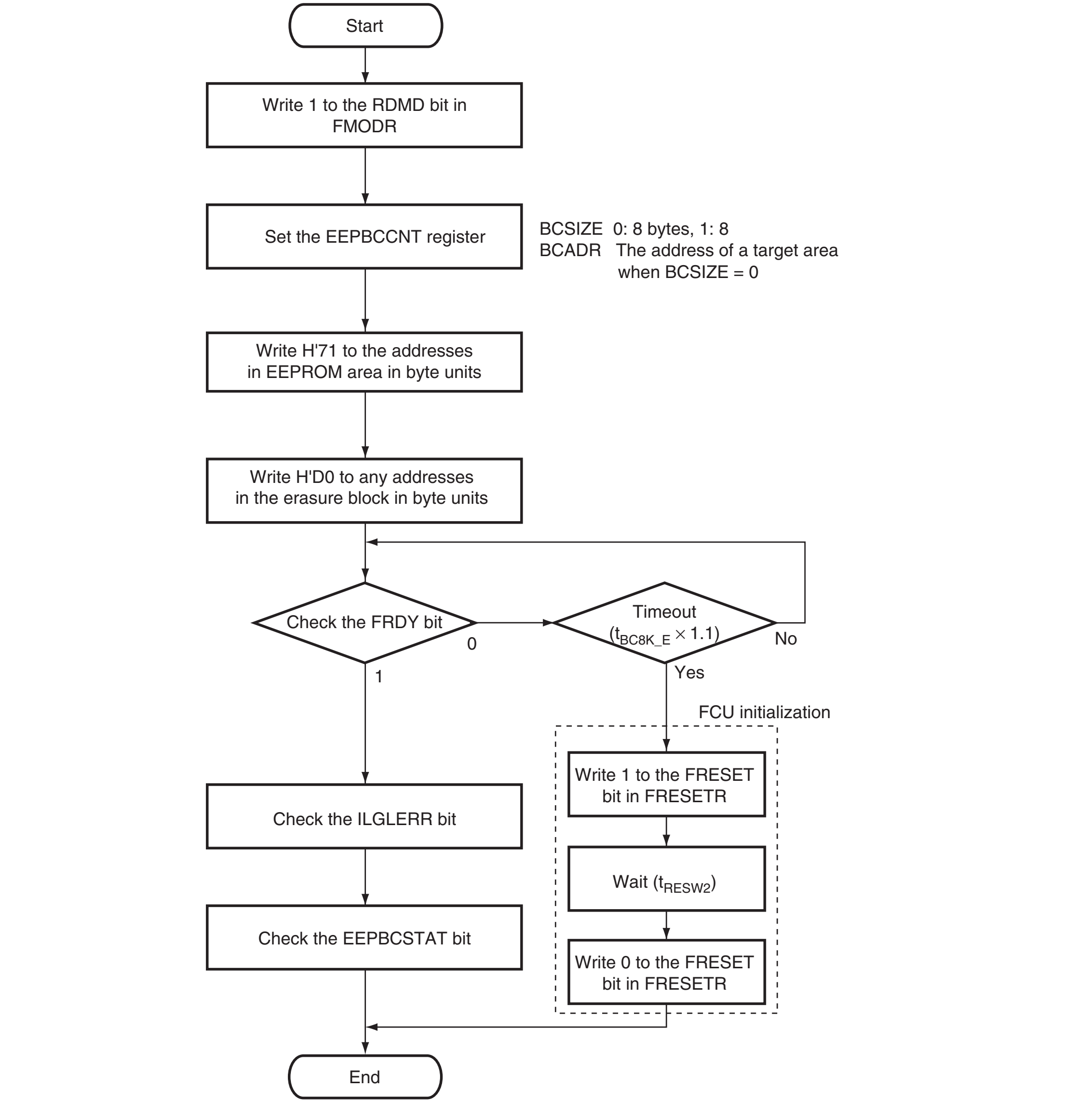
(2) Erasure

To erase the ROM, use the block erase command. The EEPROM can be erased in the same way as ROM erasure (refer to section 26, ROM). Note that the EEPROM has a programming and erasure protection function through EEPWE0 and EEPWE1. To ignore the programming and erasure protection provided by the EEPWE0 and EEPWE1 settings, set the program/erase enable bit for the target block to 1 before starting erasure.

(3) Checking of the Erased State

Since reading the EEPROM erased by the CPU results in undefined values, the blank check command should be used to check the erased state of the EEPROM. To make the blank check command available for use, set the FRDMD bit in FMODR to 1 to enable the command first, and then specify the size and start address of a target area via the EEPBCCNT register. When the BCSIZE bit of the EEPBCCNT register is set to 1, a check can be performed on the entire erased block (8 Kbytes) specified in the second cycle of the command. When the BCSIZE bit is set to 0, a check can be performed on an 8-byte area starting from the address obtained by summing the start address of the erased area specified in the second cycle of the command and the value held by the EEPBCCNT register. In the first cycle of the command, a value of H'71 is written in byte into an address of the EEPROM. In the second cycle, once a value of H'D0 is written into a specified address included in the target area, the FCU starts the blank check on the EEPROM. It can be checked whether or not the check is complete via the FRDY bit in the FSTATR0. After the blank check is complete, it can be checked whether the target area is erased or filled with 0s and/or 1s via the BCST bit of the EEPBCSTAT register.

Figure 28.7 shows the procedure of the EEPROM blank check.


Start
Write 1 to the RDMD bit in FMODR
Set the EEPBCCNT register
BCSIZE 0: 8 bytes, 1: 8
BCADR The address of a target area when BCSIZE = 0
Write H'71 to the addresses in EEPROM area in byte units
Write H'D0 to any addresses in the erasure block in byte units
Check the FRDY bit
0
1
Timeout (t_{BC8K_E} × 1.1)
No
Yes
FCU initialization
Write 1 to the FRESET bit in FRESETR
Wait (t_{RESW2})
Write 0 to the FRESET bit in FRESETR
Check the ILGLERR bit
Check the EEPBCSTAT bit
End

Notes: t_{BC8K_E}: Time required for blank check (see section 34, Electrical Characteristics) .
t_{RESW2}: Reset pulse width during programming and erasure (see section 34, Electrical Characteristics).

Figure 28.7 Procedure of the EEPROM Blank Check

28.7 Protection

There are three types of EEPROM programming/erasure protection: hardware, software, and error protection.

28.7.1 Hardware Protection

The hardware protection function disables EEPROM programming and erasure according to the mode pin settings in this LSI.

While the on-chip ROM is disabled, EEPROM programming, erasing, and reading are disabled. For the operating modes set through the mode pins of this LSI, refer to section 3, Operating Modes.

28.7.2 Software Protection

The software protection function disables EEPROM programming and erasure according to the control register settings. If an attempt is made to issue a programming or erasing command to the EEPROM against software protection, the FCU detects an error and enters command-locked state.

(1) Protection through FENTRYR

When the FENTRYD bit in FENTRYR is 0, the FCU does not accept commands for the EEPROM, so EEPROM programming and erasure are disabled. If an attempt is made to issue an FCU command for the EEPROM while the FENTRYD bit is 0, the FCU detects an illegal command error and enters command-locked state (see section 28.7.3, Error Protection).

(2) Protection through EEPWE0 and EEPWE1

When the DBWEi (i = 00 to 15) bit in EEPWE0 or EEPWE1 is 0, programming and erasure of block DBi in the data MAT is disabled. If an attempt is made to program or erasure block DBi while the DBWEi bit is 0, the FCU detects a program/erase protect error and enters command-locked state (see section 28.7.3, Error Protection).

28.7.3 Error Protection

The error protection function detects an illegal FCU command issued, an illegal access, or an FCU malfunction, and disables FCU command acceptance (command-locked state). While the FCU is in command-locked state, the EEPROM cannot be programmed or erased. To cancel command-locked state, issue a status register clear command while FASTAT is H'10.

While the CMDLKIE bit in FAEINT is 1, a flash interface error (FIFE) interrupt is generated if the FCU enters command-locked state (the CMDLK bit in FASTAT becomes 1). While an EEPROM-related interrupt enable bit (EEPAEIE, EEPIFEIE, EEPRPEIE, or EEPWPEIE) in FAEINT is 1, an FIFE interrupt is generated if the corresponding status bit (EEPAE, EEPIFE, EEPRPE, or EEPWPE) in FASTAT becomes 1.

Table 28.9 shows the error protection types for the EEPROM and the status bit values (the ILGLERR, ERSERR, and PRGERR bits in FSTATR0 and the EEPAE, EEPIFE, EEPRPE, and EEPWPE bits in FASTAT) after each error detection. For the error protection types used in common by the ROM and EEPROM (FENTRYR setting error, most of illegal command errors, erasing error, programming error, FCU error, and FCU RAM ECC error), refer to section 26.8.3, Error Protection. If the FCU enters command-locked state due to a command other than a suspend command issued during programming or erasure processing, the FCU continues programming or erasing the EEPROM. In this state, the P/E suspend command cannot suspend programming or erasure. If a command is issued in command-locked state, the ILGLERR bit becomes 1 and the other bits retain the values set due to the previous error detection.

Table 28.9 Error Protection Types (for EEPROM only)

Error	Description	ILGLERR	ERSERR	PRGERR	EEPAE	EEPIFE	EEPRPE	EEPWPE
Illegal command error	The value specified in the second cycle of a program command is neither H'04 nor H'40.	1	0	0	0	0	0	0
	A lock bit program command has been issued to an area in the EEPROM while the FENTRYD bit of FENTRYR register is set to 1.	1	0	0	0	0	0	0
EEPROM access error	A read access command has been issued to the EEPROM area while FENTRYD = 1 in FENTRYR in EEPROM P/E normal mode.	1	0	0	1	0	0	0
	A write access command has been issued to the EEPROM area while FENTRYD = 0.	1	0	0	1	0	0	0
	An access command has been issued to the EEPROM area while one of the bits, FENTRY6 to FENTRY0, in FENTRYR is 1.	1	0	0	1	0	0	0
EEPROM instruction fetch error	An instruction fetch has been made in the EEPROM area.	1	0	0	0	1	0	0
EEPROM read protect error	A read access command has been issued to the EEPROM area protected against reading through EEPRE0 and EEPRE1.	1	0	0	0	0	1	0
EEPROM program/erase protect error	A program command or block erase command has been issued to the EEPROM area protected against programming and erasure through EEPWE0 and EEPWE1.	1	0	0	0	0	0	1

28.8 Product Information MAT

The product information MAT stores the device name, device revision number, and embedded program revision number information in ASCII code. The embedded program is stored in the reset-start MAT used in boot mode and user boot mode (refer to section 26.4, Overview of ROM-Related Modes). Table 28.10 shows the addresses to store the information and an example of information data. In the product information MAT (H'80100000 to H'80100078), the addresses not shown in this table are reserved areas. Undefined data will be read from the reserved areas.

Table 28.10 Data Stored in Product Information MAT

Information	Address	Example of Data
Device name	H'80100000 to H'8010000F	H'52354637323536375220202020202020 = R5F72567R
Device revision number	H'80100010 to H'80100011	H'3031 = 01
Embedded program revision number	H'80100020 to H'80100022	H'313030 = 100 (1.00)

28.9 Usage Notes

(1) Protection of Data MAT Immediately after a Reset

As the initial value of EEPRE0, EEPRE1, EEPWE0, and EEPWE1 is H'0000, data MAT programming, erasure, and reading are disabled immediately after a reset. To read data from the data MAT, set EEPRE0 and EEPRE1 appropriately before accessing the data MAT. To program or erase the data MAT, set EEPWE0 and EEPWE1 appropriately before issuing an FCU command for programming or erasure. If an attempt is made to read, program, or erase the data MAT without setting the registers, the FCU detects an error and enters command-locked state.

(2) State in which AUD Operation Is Disabled and Interrupts Are Ignored

In the following modes or period, the AUD is in module standby mode and cannot operate. The NMI or maskable interrupt requests are ignored.

- Boot mode
- The program in the embedded program stored MAT is being executed immediately after the LSI is started in user boot mode

(3) Programming-/Erasure-Suspended Area

The data stored in the programming-suspended or erasure-suspended area is undetermined. To avoid malfunction due to undefined read data, ensure that no data is read from the programming-suspended or erasure-suspended area.

(4) Compatibility with Programming/Erasing Program of Conventional F-ZTAT SH Microcontrollers

The flash memory programming/erasing program used for conventional F-ZTAT SH microcontrollers does not work with this LSI.

(5) Reset during Programming or Erasure

To reset the FCU by setting the FRESET bit in the FRESETR register during programming or erasure, hold the FCU in the reset state for a period of t_{RESW2} (see section 34, Electrical Characteristics). Since a high voltage is applied to the EEPROM during programming and erasure, the FCU has to be held in the reset state long enough to ensure that the voltage applied to the memory unit has dropped. Do not read from the EEPROM while the FCU is in the reset state.

When a power-on reset is generated by asserting the $\overline{\text{RES}}$ pin during programming or erasure of the flash memory, hold the reset state for a period of t_{RESW2} (see section 34, Electrical Characteristics). In a power-on reset, not only does the voltage applied to the memory unit have to drop, but the power supply for the EEPROM and its internal circuitry also have to be initialized. Thus, the reset state must be maintained over a longer period than in the case of resetting the FCU.

While programming or erasure is performed, do not generate an internal reset caused by WDT counter overflow. A reset caused by WDT cannot ensure a sufficient time required for voltage drop for the memory unit, initialization of the power supply for the EEPROM, or initialization of its internal circuit.

When either a power-on reset by asserting the $\overline{\text{RES}}$ pin, or an FCU reset by setting the FRESET bit in the FRESETR register, is executed during programming or erasure, the whole data in the programming or erasure area becomes undefined.

No verification function is provided to check program/erase state of the area where the data is undefined by suspend of program/erase (e.g., reset input, power-supply interruption). Therefore, if the undefined area should be used again, make sure to completely erase data before usage.

(6) Prohibition of Additional Programming

One area cannot be programmed twice in succession. To program an area that has already been programmed, be sure to erase the area before reprogramming.

(7) Writing to/Erasing the Product Information MAT

Since the product information MAT is read only, no write/erase operation can be performed. When write/erase operation is performed with the EEPSEL bit in the EEPMAT register set to 1, the data MAT is written to or erased without any error occurring such as EEPROM access violation. Therefore, never write to or erase the product MAT.

(8) Suspension by Programming or Erasure Suspension

When programming or erasure is suspended by a programming or an erasure suspend command, the programming or erasure must be completed by a resume command.

(9) Power off during Programming or Erasure

Do not switch the power off during programming or erasure. As a high voltage is applied to the ROM during programming and erasure, a certain period is required for this voltage to fall.

Thus, to allow for cases where switching the power off during programming or erasure cannot be avoided, design the system so that at least the V_{CC} holding time at PV_{CC} shutdown (t_{VCCH}) is secured by assertion of the $\overline{\text{HSTBY}}$ signal causing a transition to the hardware standby state if the power is switched off. For details, see section 34, Electrical Characteristics.

(10) Prohibition of Clearing FRDCLE Bit to 0

Whenever the FRDTCT bit in FSTATR1 is set to 1, the FCU has to enter command-locked state, because the FCU command operation cannot be guaranteed. Thus, do not clear the FRDCLE bit in FRAMECCR to 0.

Section 29 ROM Cache (ROMC)

The ROM cache is designed to cache the instructions and data stored in the ROM, permitting high-speed access to these instructions and data.

29.1 Features

- Configuration: Separate caches for instructions and data
- Prefetch cache: 8-line, 4-way set associative, LRU method
- Prefetch-miss cache: 4-line, fully associative, LRU method
- Data cache: 4-line, fully associative, LRU method, automatic line invalidation
- Line size: 16 bytes (128 bits)
- Hardware prefetching: Instructions are read from the ROM and stored in the prefetch cache prior to instruction fetching by the CPU

29.2 Configuration

The ROM-cache module has separate units for instructions and data. The instruction caches consist of a prefetch cache (for instruction read-ahead) and prefetch-miss cache (where data read from the ROM are stored when neither the prefetch cache nor the prefetch-miss cache currently contains the desired data). The data cache is for the storage of data.

Figure 29.1 shows the configuration of the caches. The prefetch cache is 8-line and 4-way set associative, while both the prefetch-miss cache and the data cache are 4-line full associative. All of the caches have 16-byte lines, and the LRU (Least Recently Used) method is the principle of the line-replacement algorithm. For detailed descriptions of the line-replacement policies for the individual caches, refer to section 29.4.1, Data Cache Lookup.

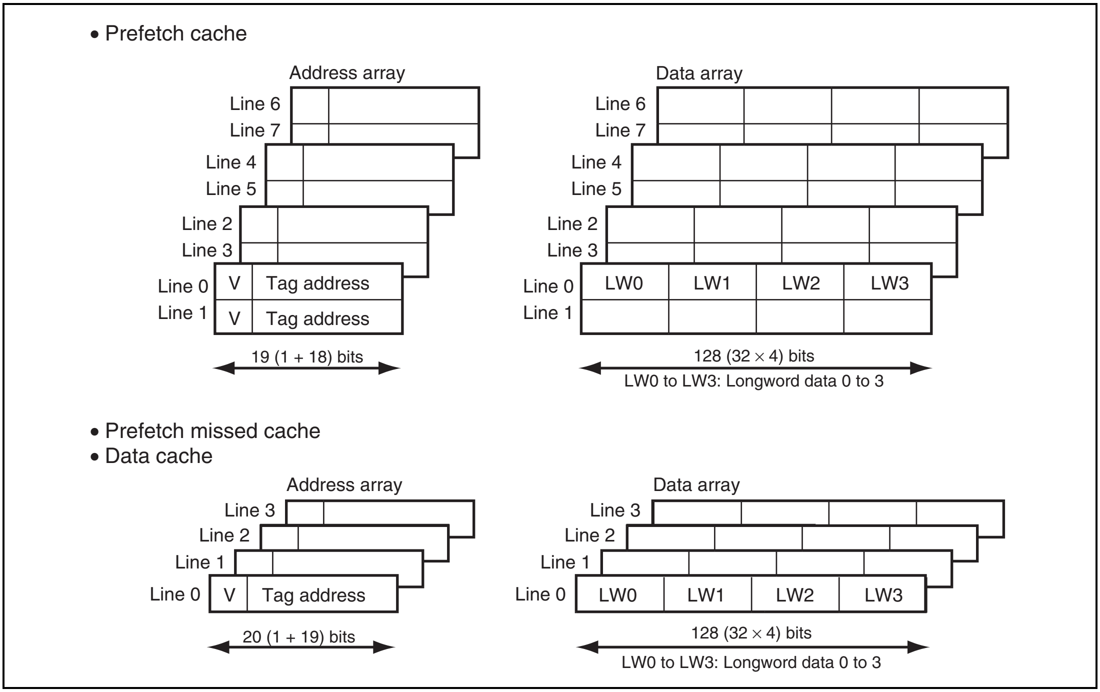


Figure 29.1 Cache Configuration

(1) Address Arrays

The address array of the prefetch cache consists of eight lines, lines 0 to 7. The address arrays of the data and prefetch-miss caches consist of four lines, lines 0 to 3. Each line is composed of a V bit and a Tag address.

The V bit indicates whether or not the line data is valid; the data line is valid when the V bit is 1 and invalid when the V bit is 0. When the target address in the ROM is identified as already cached, the corresponding line is invalidated by setting the V bit to 0.

The Tag addresses hold the addresses for reference in the cache lookup process. Each tag consists of 18 bits (corresponding to bits 22 to 5 of the access address in the ROM) in the prefetch cache and 19 bits (bits 22 to 4 of the access address) in the prefetch-miss cache and the data cache. Bits 31 to 23 of the access address are used on the address bus to identify the memory space, and are thus not relevant to the cache lookup process.

The V bits for the instruction caches or data cache are initialized to 0 by writing a 1 to the corresponding flush bit in the ROM cache control register. The whole cache is initialized to 0 by writing a 1 to the main flush bit in the ROM cache control register and on entry to the reset or standby state.

(2) Data Array

The data arrays of the data and prefetch-miss caches consist of four lines, lines 0 to 3. Each line is 16 bytes long and is divided into four longwords, LW0 to LW3. The 16 bytes of instructions or data held by a single line is the smallest unit of caching.

Data stored in the data array are undefined after a reset.

29.3 Register Descriptions

The ROM cache has the following registers. These registers can only be accessed as longwords. Table 29.1 shows the configuration of the cache-related registers.

Table 29.1 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
ROM cache control register	RCCR	R/W	H'0000 0001	H'FFFC 1400	32
ROM cache control register 2	RCCR2	R/W	H'0000 00F5	H'FFFC 1408	32

29.3.1 ROM Cache Control Register (RCCR)

The RCE bit of RCCR specifies enabling or disabling of the ROM cache. RCCR also contains the RCF bit, which can be used to invalidate all lines in the ROM cache, the RCFI bit, which can be used to invalidate all lines of the instruction caches (the prefetch and prefetch-miss caches), the RCFD bit, which can be used to invalidate all lines in the data cache.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	RCF	RCFI	RCFD	RCE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	RCF	0	R/W	ROM Cache Flush Writing a 1 to this bit clears the V bits of all lines in the ROM cache to 0 (flushes the cache). This bit is read as 0. 0: Does not clear the V bits in the ROM cache lines. 1: Clears the V bits in the ROM cache lines. [Clearing condition] • Reset/standby [Setting condition] • Writing a 1.
2	RCFI	0	R/W	Instruction Cache Flush Writing a 1 to this bit clears the V bits of all lines in the prefetch and prefetch-miss-caches to 0 (flushes the prefetch and prefetch-miss caches). This bit is read as 0. 0: Does not clear the V bits in the instruction cache lines. 1: Clears the V bits in the instruction cache lines. [Clearing condition] • Reset/standby [Setting condition] • Writing a 1.
1	RCFD	0	R/W	Data Cache Flush Writing a 1 to this bit clears the V bits of all lines in the data cache to 0 (flushes the data cache). This bit is read as 0. 0: Does not clear the V bits in the data cache lines. 1: Clears the V bits in the data cache lines. [Clearing condition] • Reset/standby [Setting condition] • Writing a 1.

Bit	Bit Name	Initial Value	R/W	Description
0	RCE	1	R/W	ROM Cache Enable Specifies usage or non-usage of ROM caching. 0: The ROM cache function is not used. 1: The ROM cache function is used. [Clearing condition] • Writing a 0. [Setting conditions] • Reset/standby • Writing a 1.

29.3.2 ROM Cache Control Register 2 (RCCR2)

The PCE2 and PCE0 bits of RCCR2 specify enabling or disabling the prefetch-miss cache and data cache, respectively. The PFE bit specifies enabling or disabling the prefetch cache.

With regard to the prefetching function, the PFECF, PFENB, and PFECB bits are used to specify prefetching of consecutive instructions, prefetching from the destinations of unconditional branches, and prefetching from the destinations of conditional branches, respectively.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	PFECB	PFENB	PFECF	PFE	-	PCE2	-	PCE0
Initial value:	0	0	0	0	0	0	0	0	1	1	1	1	0	1	0	1
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R/W	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	PFECB	1	R/W	Conditional Branch Prefetch Enabled Specifies whether or not data is to be prefetched from the destinations of conditional branches. 0: No prefetching from the destinations of conditional branches. 1: Prefetching from the destinations of conditional branches. [Clearing conditions] • Reset/standby • Writing a 0. [Setting condition] • Writing a 1.
6	PFENB	1	R/W	Unconditional Branch Prefetch Enable Specifies whether or not data is to be prefetched from the destinations of unconditional branches. 0: No prefetching from the destinations of unconditional branches. 1: Prefetching from the destinations of unconditional branches. [Clearing conditions] • Reset/standby • Writing a 0. [Setting condition] • Writing a 1.

Bit	Bit Name	Initial Value	R/W	Description
5	PFECF	1	R/W	Consecutive Prefetch Enable Specifies whether or not prefetching is applied to instructions for consecutive execution (consecutive instructions). 0: Consecutive instructions are not prefetched. 1: Consecutive instructions are prefetched. [Clearing conditions] • Reset/standby • Writing a 0. [Setting condition] • Writing a 1.
4	PFE	1	R/W	Prefetch Cache Enable Specifies usage or non-usage of the prefetch cache. 0: The prefetch cache function is not used. 1: The prefetch cache function is used. [Clearing conditions] • Reset/standby • Writing a 0. [Setting condition] • Writing a 1.
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	PCE2	1	R/W	Prefetch-Miss Cache Enable Specifies usage or non-usage of the prefetch-miss cache. 0: The prefetch-miss cache function is not used. 1: The prefetch-miss cache function is used. [Clearing conditions] • Reset/standby • Writing a 0. [Setting condition] • Writing a 1.

Bit	Bit Name	Initial Value	R/W	Description
1	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
0	PCE0	1	R/W	Data Cache Enable Specifies usage or non-usage of the data cache. 0: The data cache function is not used. 1: The data cache function is used. [Clearing conditions] • Reset/standby • Writing a 0. [Setting condition] • Writing a 1.

29.4 Operation

29.4.1 Data Cache Lookup

When both ROM caching and the data cache are enabled (RCCR.RCE = 1 and RCCR2.PCE0 = 1) and data is read from the ROM area, the data cache is checked to see if it holds valid target data. The tag addresses of all four lines are simultaneously fed to four comparators, each of which compares bits 22 to 4 of one tag address with the access address. If the result of comparison is a match and the compared line is valid (the V bit = 1), the cache has been "hit" and LW0 to LW3 in the corresponding line of the data array are read out. Otherwise, the result is regarded as a cache miss.

Bits 3 and 2 of the access address then indicate the target longword for output to the CPU: 00 corresponds to LW0, 01 to LW1, 10 to LW2, and 11 to LW3.

Figure 29.2 shows the concept of looking up the data cache, with line 1 being hit.

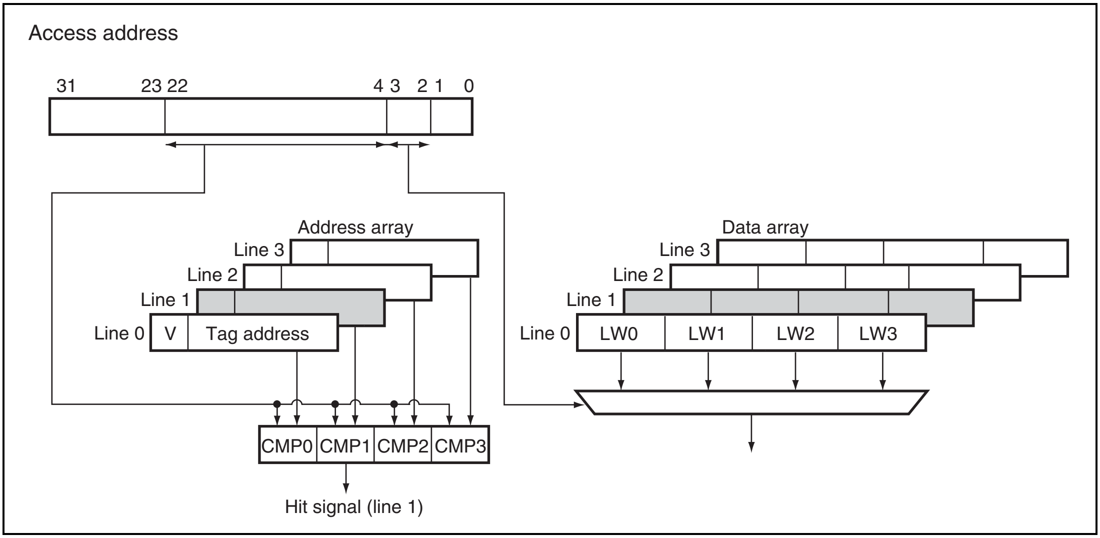


Figure 29.2 Concept of Looking up the Data Cache (with Line 1 being Hit)

If the data cache has not been hit (the cache was missed), data for the cache line is read out from the corresponding actual addresses in the ROM and bits 22 to 4 of the target address replace the tag address for the least recently used line (LRU method) of the address array. At the same time, the V bit is set and data from the target address is also output to the CPU.

29.4.2 Instruction Cache Lookup

In looking up the prefetch cache, whether the value of bit 4 of the access address is 0 or 1 determines whether the even- (0, 2, 4, or 6) or odd-numbered (1, 3, 5, or 7) lines should be checked, respectively. The tag addresses of the four selected lines are then simultaneously fed to four comparators, each of which compares bits 22 to 5 of one tag address with the access address. If the result of comparison is a match and the compared line is valid (the V bit = 1), the cache has been hit and LW0 to LW3 in the corresponding line of the data array are read out. Otherwise, the result is regarded as a cache miss.

Bits 3 and 2 of the access address then indicate the target longword for output to the CPU: 00 corresponds to LW0, 01 to LW1, 10 to LW2, and 11 to LW3.

Figure 29.3 shows the concept of looking up the prefetch cache, with line 2 being hit.

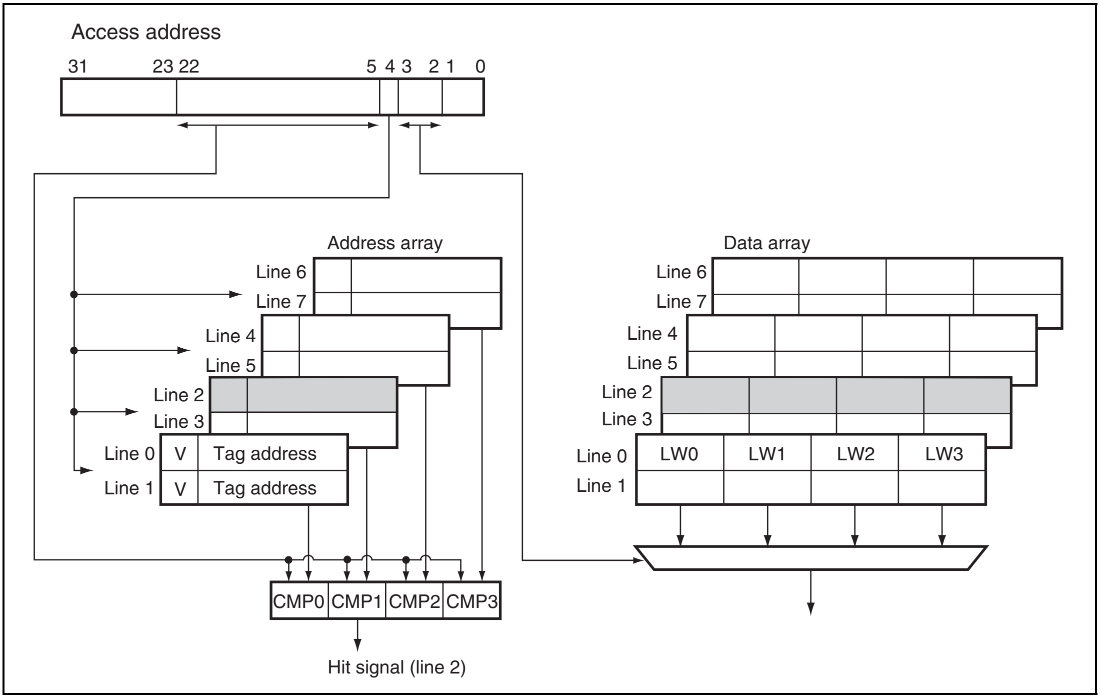


Figure 29.3 Concept of Looking up the Prefetch Cache (with Line 2 being Hit)

The prefetch-miss cache is looked up in the same way as the data cache. When a cache miss is produced by looking up both the prefetch and prefetch-miss cache, data for the cache line is read out from the corresponding actual addresses in the ROM and bits 22 to 4 of the target address replace the tag address for the least recently used line (LRU method) of the address array. At the same time, the V bit is set and the instruction from the target address is also output to the CPU.

The prefetch cache is constantly updated if hardware prefetching is enabled. Whether bit 4 of the address containing the prefetched instruction is 0 or 1 determines whether the line to be updated is the least recently used of the even- or odd-numbered lines, respectively. Updating is done by replacing the tag address of the least recently used line with bits 22 to 5 of the read address and reading the data for the corresponding line of the data array from the ROM.

29.4.3 Hardware Prefetching

The ROM cache improves the hit rate of its component caches by hardware prefetching, including fetching in anticipation of the results of branch instructions. Prefetching is intended to prevent stalling in fetching by the processor, i.e. to ensure that the cache constantly supplies the processor with instructions to execute. There are two types of prefetching: consecutive prefetching and branch prefetching.

(1) Consecutive Prefetching

Instructions that are subject to consecutive access are prefetched and placed in the prefetch cache.

(2) Branch Prefetching

In branch prefetching, branch instructions are prefetched and decoded, their target addresses are predicted, and the instructions at those addresses are prefetched. Branch prefetching is subdivided into prefetching from the targets of conditional branches and prefetching from the targets of unconditional branches.

- Conditional-branch prefetching

  On encountering a conditional branch instruction (BF, BT, BF/S, or BT/S), prefetching from the predicted destination address is performed.

- Unconditional-branch prefetching

  BRA and BSR instructions

  If unconditional-branch prefetching has been selected, prefetching always proceeds on encountering these unconditional branch instructions.

- JMP, JSR, and JSR/N instructions

  If unconditional-branch prefetching has been selected, prefetching from the destinations of branch instructions of the above type is performed when the instruction has a register index and the index is fully predictable.

The following is an example of instructions that will lead to prefetching from the destination address of the JMP instruction when consecutive-instructive and unconditional-branch prefetching have been selected (JMP):

```
MOVI20 #imm20, Rn
JMP @Rn
```

29.5 Usage Note

If a register of the ROMC is written to while the ROM is read by the DMAC or AUD-II, data may not be read correctly from the ROM. Do not write to the ROMC registers when the ROM is being accessed by the DMAC or AUD-II.

Section 30 RAM

This LSI incorporates 256-Kbyte RAM, which is connected to F (CPU instruction Fetch), M (Memory access), and I (Internal) buses. This on-chip RAM can be accessed via any of these buses independently.

Figure 30.1 shows RAM block diagrams and figure 30.2 shows RAM and bus connections.

The on-chip RAM is allocated in addresses H'FFF80000 to H'FFFBFFFF (pages 0 to 15), as shown in table 30.1.

30.1 Features

- Access

  The CPU/FPU, DMAC, A-DMAC, and AUD-II can access on-chip RAM in 8, 16, or 32 bits. Data in the on-chip RAM can be effectively used as program area or stack area data necessary for access at high speed.

  For reading from the on-chip RAM, 1 or 2 cycles should be specified by a register according to the operating frequency (PLL multiplication ratio). For writing to the on-chip RAM, 2 or 3 cycles must be specified by a register according to the operating frequency (PLL multiplication ratio).
- RAM data retention

  The data in pages 1 and 0, a 32-Kbyte area, of RAM is retained during hardware standby.
- ECC

  The ECC error correction function can be enabled or disabled by register settings. The function is initially enabled. When the ECC is enabled, correction of one erroneous bit and detection of 2 erroneous bits are possible for 32 bits of data. When the ECC function is disabled, parity error detection is available. These ECC error detection and correction and parity errors are collectively called RAM errors. Flags that indicate occurrence of these RAM errors are provided.
- Interrupts

  Whether or not an interrupt is requested upon occurrence of a RAM error can be selected by register settings.
- Ports

  Each page in the on-chip RAM has two independent read and write ports. The read port is connected to I, F, and M buses and the write port is connected to I and M buses. The F and M buses are used for accesses from the CPU. The I bus is used for accesses from other than the CPU.

- Priority

  If the same page is accessed from multiple buses simultaneously, the access is performed according to the bus priority. The bus priority is as follows: I bus (highest), M bus (middle), F bus (lowest).

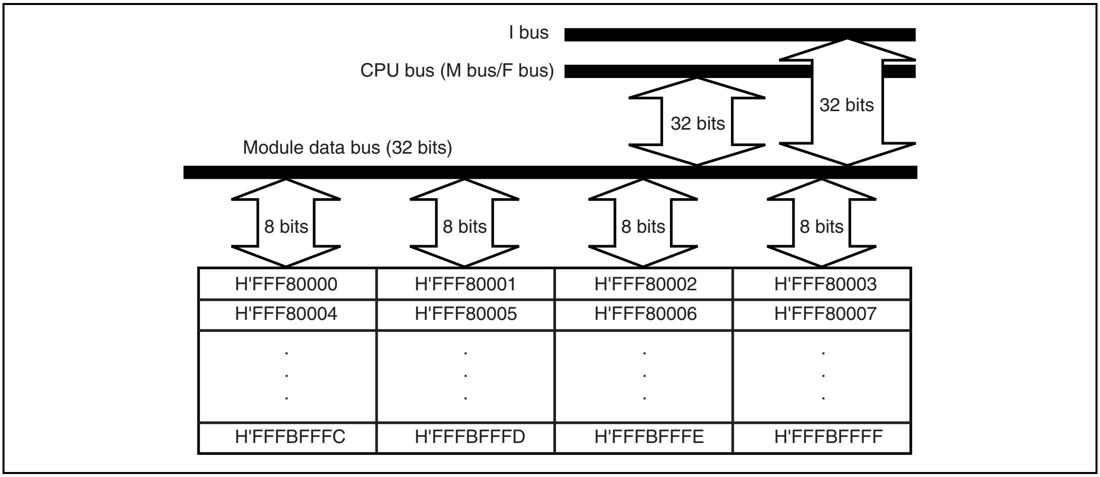


Figure 30.1 RAM Block Diagram

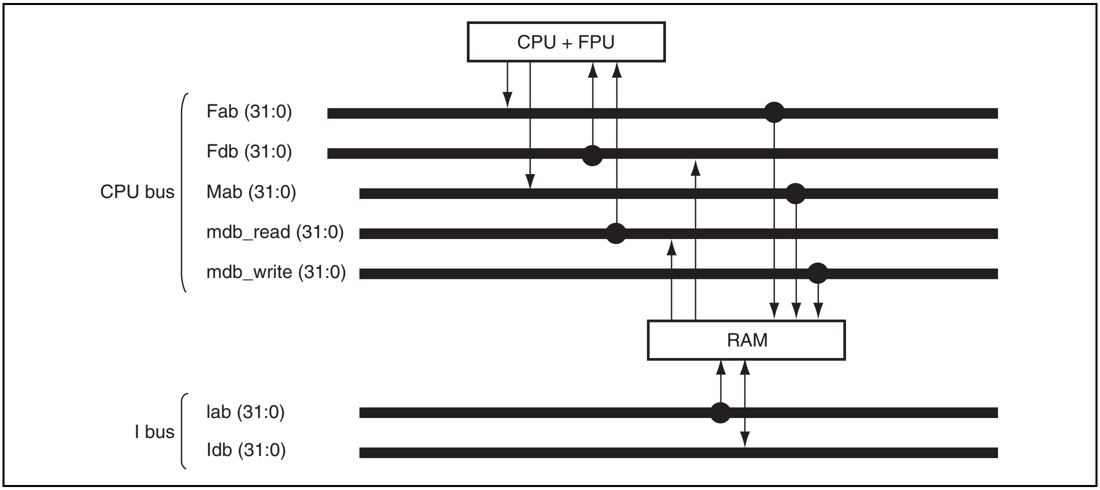


Figure 30.2 Bus Connections in RAM

Table 30.1 On-chip RAM Address Space

Page	Address
Page 0	H'FFF80000 to H'FFF83FFF
Page 1	H'FFF84000 to H'FFF87FFF
Page 2	H'FFF88000 to H'FFF8BFFF
Page 3	H'FFF8C000 to H'FFF8FFFF
Page 4	H'FFF90000 to H'FFF93FFF
Page 5	H'FFF94000 to H'FFF97FFF
Page 6	H'FFF98000 to H'FFF9BFFF
Page 7	H'FFF9C000 to H'FFF9FFFF
Page 8	H'FFFA0000 to H'FFFA3FFF
Page 9	H'FFFA4000 to H'FFFA7FFF
Page 10	H'FFFA8000 to H'FFFABFFF
Page 11	H'FFFAC000 to H'FFFAFFFF
Page 12	H'FFFB0000 to H'FFFB3FFF
Page 13	H'FFFB4000 to H'FFFB7FFF
Page 14	H'FFFB8000 to H'FFFBBFFF
Page 15	H'FFFBC000 to H'FFFBFFFF

30.2 Register Descriptions

The on-chip RAM has registers shown in table 30.2.

Table 30.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
RAM enable control register 0	RAMEN0	R/W	H'00FF	H'FFFF0800	8, (16)
RAM write enable control register 0	RAMWEN0	R/W	H'00FF	H'FFFF0802	8, (16)
RAM ECC enable control register	RAMECC	R/W	H'0000	H'FFFF0804	8, (16)
RAM error status register	RAMERR	R/W	H'00	H'FFFF0806	8
RAM error interrupt control register	RAMINT	R/W	H'00	H'FFFF0810	8
RAM access cycle set register	RAMACYC	R/W	H'0000	H'FFFF0812	8, (16)
RAM enable control register 1	RAMEN1	R/W	H'00FF	H'FFFF0840	8, (16)
RAM write enable control register 1	RAMWEN1	R/W	H'00FF	H'FFFF0842	8, (16)

30.2.1 RAM Enable Control Registers 0 and 1 (RAMEN0 and RAMEN1)

The RAM enable control registers (RAMEN0 and RAMEN1) are 16-bit readable/writable registers that enable or disable the access to the on-chip RAM. RAMEN0 and RAMEN1 are initialized to H'00FF by a reset or in the standby state. They can be written to in words, and can be read in bytes or words.

If the RAME15 to RAME0 bits corresponding to the access page are set to 1, accessing the on-chip RAM becomes enabled; while if the RAME bits are cleared to 0, the on-chip RAM cannot be accessed. In the access disabled state, an undefined data is read if the page is read or if an instruction in the page is fetched, and a write to the page is ignored. The initial values of the RAME bits are 1.

To rewrite the RAME15 to RAME0 bits in RAMEN0, word size data with upper byte as H'96 and lower byte as write data must be written. Similarly, to rewrite the RAME15 to RAME0 bits in RAMEN1, word size data with upper byte as H'95 and lower byte as write data must be written.

When the upper byte (bits 15 to 8) of RAMEN0 or RAMEN1 is read, H'00 is always read.

Any instructions to access the on-chip RAM must not be placed immediately after any instructions to write to RAMEN0 or RAMEN1. Otherwise, correct access to the on-chip RAM cannot be guaranteed.

To rewrite the RAME15 to RAME0 bits in RAMEN0, execute an instruction to read from RAMEN0 together with five or more NOP instructions immediately after the instruction to write to RAMEN0. To rewrite the RAME15 to RAME0 bits in RAMEN1, execute instructions for RAMEN1 in the same way as for RAMEN0.

- RAMEN0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RN0KEY[7:0]								RAME 7	RAME 6	RAME 5	RAME 4	RAME 3	RAME 2	RAME 1	RAME 0
Initial value:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
R/W:	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: To avoid erroneous rewriting, the way of writing data to this register is different from that to other general registers. For details, see section 30.2.7, Notes on Register Access.

Bit	Bit Name	Initial Value	R/W	Descriptions
15 to 8	RN0KEY[7:0]	H'00	R/(W)	These bits enable or disable write to the RAME bit. H'96: Enable write to bits RAME7 to RAME0. The write data is not retained and these bits are always read as H'00. Other than H'96: Disable write to bits RAME7 to RAME0.
7	RAME7	1	R/W	RAM Enable 7 Enables or disables access to page 7 in the on-chip RAM. 0: Disables access to page 7 in the on-chip RAM 1: Enables access to page 7 in the on-chip RAM [Clearing condition] • Writing 0 (H'96 is written to the upper byte simultaneously.) [Setting conditions] • Reset or standby • Writing 1 (H'96 is written to the upper byte simultaneously.)

Bit	Bit Name	Initial Value	R/W	Descriptions
6	RAME6	1	R/W	RAM Enable 6 Enables or disables access to page 6 in the on-chip RAM. 0: Disables access to page 6 in the on-chip RAM 1: Enables access to page 6 in the on-chip RAM [Clearing condition] • Writing 0 (H'96 is written to the upper byte simultaneously.) [Setting conditions] • Reset or standby • Writing 1 (H'96 is written to the upper byte simultaneously.)
5	RAME5	1	R/W	RAM Enable 5 Enables or disables access to page 5 in the on-chip RAM. 0: Disables access to page 5 in the on-chip RAM 1: Enables access to page 5 in the on-chip RAM [Clearing condition] • Writing 0 (H'96 is written to the upper byte simultaneously.) [Setting conditions] • Reset or standby • Writing 1 (H'96 is written to the upper byte simultaneously.)

Bit	Bit Name	Initial Value	R/W	Descriptions
4	RAME4	1	R/W	RAM Enable 4 Enables or disables access to page 4 in the on-chip RAM. 0: Disables access to page 4 in the on-chip RAM 1: Enables access to page 4 in the on-chip RAM [Clearing condition] • Writing 0 (H'96 is written to the upper byte simultaneously.) [Setting conditions] • Reset or standby • Writing 1 (H'96 is written to the upper byte simultaneously.)
3	RAME3	1	R/W	RAM Enable 3 Enables or disables access to page 3 in the on-chip RAM. 0: Disables access to page 3 in the on-chip RAM 1: Enables access to page 3 in the on-chip RAM [Clearing condition] • Writing 0 (H'96 is written to the upper byte simultaneously.) [Setting conditions] • Reset or standby • Writing 1 (H'96 is written to the upper byte simultaneously.)

Bit	Bit Name	Initial Value	R/W	Descriptions
2	RAME2	1	R/W	RAM Enable 2 Enables or disables access to page 2 in the on-chip RAM. 0: Disables access to page 2 in the on-chip RAM 1: Enables access to page 2 in the on-chip RAM [Clearing condition] • Writing 0 (H'96 is written to the upper byte simultaneously.) [Setting conditions] • Reset or standby • Writing 1 (H'96 is written to the upper byte simultaneously.)
1	RAME1	1	R/W	RAM Enable 1 Enables or disables access to page 1 in the on-chip RAM. 0: Disables access to page 1 in the on-chip RAM 1: Enables access to page 1 in the on-chip RAM [Clearing condition] • Writing 0 (H'96 is written to the upper byte simultaneously.) [Setting conditions] • Reset or standby • Writing 1 (H'96 is written to the upper byte simultaneously.)

Bit	Bit Name	Initial Value	R/W	Descriptions
0	RAME0	1	R/W	RAM Enable 0 Enables or disables access to page 0 in the on-chip RAM. 0: Disables access to page 0 in the on-chip RAM 1: Enables access to page 0 in the on-chip RAM [Clearing condition] • Writing 0 (H'96 is written to the upper byte simultaneously.) [Setting conditions] • Reset or standby • Writing 1 (H'96 is written to the upper byte simultaneously.)

- RAMEN1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RN1KEY[7:0]								RAME 15	RAME 14	RAME 13	RAME 12	RAME 11	RAME 10	RAME 9	RAME 8
Initial value:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
R/W:	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: To avoid erroneous rewriting, the way of writing data to this register is different from that to other general registers. For details, see section 30.2.7, Notes on Register Access.

Bit	Bit Name	Initial Value	R/W	Descriptions
15 to 8	RN1KEY[7:0]	H'00	R/(W)	These bits enable or disable write to the RAME bit. H'95: Enable write to bits RAME15 to RAME8. The write data is not retained and these bits are always read as H'00. Other than H'95: Disable write to bits RAME15 to RAME8.
7	RAME15	1	R/W	RAM Enable 15 Enables or disables access to page 15 in the on-chip RAM. 0: Disables access to page 15 in the on-chip RAM 1: Enables access to page 15 in the on-chip RAM [Clearing condition] • Writing 0 (H'95 is written to the upper byte simultaneously.) [Setting conditions] • Reset or standby • Writing 1 (H'95 is written to the upper byte simultaneously.)

Bit	Bit Name	Initial Value	R/W	Descriptions
6	RAME14	1	R/W	RAM Enable 14 Enables or disables access to page 14 in the on-chip RAM. 0: Disables access to page 14 in the on-chip RAM 1: Enables access to page 14 in the on-chip RAM [Clearing condition] • Writing 0 (H'95 is written to the upper byte simultaneously.) [Setting conditions] • Reset or standby • Writing 1 (H'95 is written to the upper byte simultaneously.)
5	RAME13	1	R/W	RAM Enable 13 Enables or disables access to page 13 in the on-chip RAM. 0: Disables access to page 13 in the on-chip RAM 1: Enables access to page 13 in the on-chip RAM [Clearing condition] • Writing 0 (H'95 is written to the upper byte simultaneously.) [Setting conditions] • Reset or standby • Writing 1 (H'95 is written to the upper byte simultaneously.)

Bit	Bit Name	Initial Value	R/W	Descriptions
4	RAME12	1	R/W	RAM Enable 12 Enables or disables access to page 12 in the on-chip RAM. 0: Disables access to page 12 in the on-chip RAM 1: Enables access to page 12 in the on-chip RAM [Clearing condition] • Writing 0 (H'95 is written to the upper byte simultaneously.) [Setting conditions] • Reset or standby • Writing 1 (H'95 is written to the upper byte simultaneously.)
3	RAME11	1	R/W	RAM Enable 11 Enables or disables access to page 11 in the on-chip RAM. 0: Disables access to page 11 in the on-chip RAM 1: Enables access to page 11 in the on-chip RAM [Clearing condition] • Writing 0 (H'95 is written to the upper byte simultaneously.) [Setting conditions] • Reset or standby • Writing 1 (H'95 is written to the upper byte simultaneously.)

Bit	Bit Name	Initial Value	R/W	Descriptions
2	RAME10	1	R/W	RAM Enable 10 Enables or disables access to page 10 in the on-chip RAM. 0: Disables access to page 10 in the on-chip RAM 1: Enables access to page 10 in the on-chip RAM [Clearing condition] • Writing 0 (H'95 is written to the upper byte simultaneously.) [Setting conditions] • Reset or standby • Writing 1 (H'95 is written to the upper byte simultaneously.)
1	RAME9	1	R/W	RAM Enable 9 Enables or disables access to page 9 in the on-chip RAM. 0: Disables access to page 9 in the on-chip RAM 1: Enables access to page 9 in the on-chip RAM [Clearing condition] • Writing 0 (H'95 is written to the upper byte simultaneously.) [Setting conditions] • Reset or standby • Writing 1 (H'95 is written to the upper byte simultaneously.)

Bit	Bit Name	Initial Value	R/W	Descriptions
0	RAME8	1	R/W	RAM Enable 8 Enables or disables access to page 8 in the on-chip RAM. 0: Disables access to page 8 in the on-chip RAM 1: Enables access to page 8 in the on-chip RAM [Clearing condition] • Writing 0 (H'95 is written to the upper byte simultaneously.) [Setting conditions] • Reset or standby • Writing 1 (H'95 is written to the upper byte simultaneously.)

30.2.2 RAM Write Enable Control Registers 0 and 1 (RAMWEN0 and RAMWEN1)

The RAM write enable control registers (RAMWEN0 and RAMWEN1) are 16-bit readable/writable registers that enable or disable the access to the on-chip RAM. RAMWEN0 and RAMWEN1 are initialized to H'00FF by a reset or in the standby state. They can be written to in words, and can be read in bytes or words.

If the RAMWE15 to RAMWE0 bits corresponding to the page to be accessed are set to 1, writing to the on-chip RAM becomes enabled; while if the RAMWE bits are cleared to 0, the on-chip RAM cannot be written to. In the access disabled state, a write to the page is ignored. The initial values of the RAMWE bits are 1.

To rewrite the RAMWE15 to RAMWE0 bits in RAMWEN0, word size data with upper byte as H'69 and lower byte as write data must be written. Similarly, to rewrite the RAMWE15 to RAMWE0 bits in RAMWEN1, word size data with upper byte as H'68 and lower byte as write data must be written.

When the upper byte (bits 15 to 8) of RAMWEN0 or RAMWEN1 is read, the read value is always H'00.

Any instructions to access the on-chip RAM must not be placed immediately after any instructions to write to RAMWEN0 or RAMWEN1. Otherwise, correct access to the on-chip RAM cannot be guaranteed.

To rewrite the RAMWE15 to RAMWE0 bits in RAMWEN0, execute an instruction to read from RAMWEN0 together with five or more NOP instructions immediately after an instruction to write to RAMWEN0. To rewrite the RAMWE15 to RAMWE0 bits in RAMWEN1, execute instructions for RAMWEN1 in the same way as for RAMWEN0.

- RAMWEN0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RWN0KEY[7:0]								RAM WE7	RAM WE6	RAM WE5	RAM WE4	RAM WE3	RAM WE2	RAM WE1	RAM WE0
Initial value:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
R/W:	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: To avoid erroneous rewriting, the way of writing data to this register is different from that to other general registers. For details, see section 30.2.7, Notes on Register Access.

Bit	Bit Name	Initial Value	R/W	Descriptions
15 to 8	RWN0KEY [7:0]	H'00	R/(W)	These bits enable or disable write to the RAMWE bit. H'69: Enable write to bits RAMWE7 to RAMWE0. The write data is not retained and these bits are always read as H'00. Other than H'69: Disable write to bits RAMWE7 to RAMWE0.
7	RAMWE7	1	R/W	RAM Write Enable 7 Enables or disables write to page 7 in the on-chip RAM. 0: Disables write to page 7 in the on-chip RAM 1: Enables write to page 7 in the on-chip RAM. [Clearing condition] • Writing 0 (H'69 is written to the upper byte simultaneously.) [Setting conditions] • Reset or standby • Writing 1 (H'69 is written to the upper byte simultaneously.)
6	RAMWE6	1	R/W	RAM Write Enable 6 Enables or disables write to page 6 in the on-chip RAM. 0: Disables write to page 6 in the on-chip RAM 1: Enables write to page 6 in the on-chip RAM [Clearing condition] • Writing 0 (H'69 is written to the upper byte simultaneously.) [Setting conditions] • Reset or standby • Writing 1 (H'69 is written to the upper byte simultaneously.)

Bit	Bit Name	Initial Value	R/W	Descriptions
5	RAMWE5	1	R/W	RAM Write Enable 5 Enables or disables write to page 5 in the on-chip RAM. 0: Disables write to page 5 in the on-chip RAM 1: Enables write to page 5 in the on-chip RAM [Clearing condition] • Writing 0 (H'69 is written to the upper byte simultaneously.) [Setting conditions] • Reset or standby • Writing 1 (H'69 is written to the upper byte simultaneously.)
4	RAMWE4	1	R/W	RAM Write Enable 4 Enables or disables write to page 4 in the on-chip RAM. 0: Disables write to page 4 in the on-chip RAM 1: Enables write to page 4 in the on-chip RAM [Clearing condition] • Writing 0 (H'69 is written to the upper byte simultaneously.) [Setting conditions] • Reset or standby • Writing 1 (H'69 is written to the upper byte simultaneously.)

Bit	Bit Name	Initial Value	R/W	Descriptions
3	RAMWE3	1	R/W	RAM Write Enable 3 Enables or disables condition for write to page 3 in the on-chip RAM. 0: Disables write to page 3 in the on-chip RAM 1: Enables write to page 3 in the on-chip RAM [Clearing condition] • Writing 0 (H'69 is written to the upper byte simultaneously.) [Setting conditions] • Reset or standby • Writing 1 (H'69 is written to the upper byte simultaneously.)
2	RAMWE2	1	R/W	RAM Write Enable 2 Enables or disables write to page 2 in the on-chip RAM. 0: Disables write to page 2 in the on-chip RAM 1: Enables write to page 2 in the on-chip RAM [Clearing condition] • Writing 0 (H'69 is written to the upper byte simultaneously.) [Setting conditions] • Reset or standby • Writing 1 (H'69 is written to the upper byte simultaneously.)

Bit	Bit Name	Initial Value	R/W	Descriptions
1	RAMWE1	1	R/W	RAM Write Enable 1 Enables or disables write to page 1 in the on-chip RAM. 0: Disables write to page 1 in the on-chip RAM 1: Enables write to page 1 in the on-chip RAM [Clearing condition] • Writing 0 (H'69 is written to the upper byte simultaneously.) [Setting conditions] • Reset or standby • Writing 1 (H'69 is written to the upper byte simultaneously.)
0	RAMWE0	1	R/W	RAM Write Enable 0 Enables or disables write to page 0 in the on-chip RAM. 0: Disables write to page 0 in the on-chip RAM 1: Enables write to page 0 in the on-chip RAM [Clearing condition] • Writing 0 (H'69 is written to the upper byte simultaneously.) [Setting conditions] • Reset or standby • Writing 1 (H'69 is written to the upper byte simultaneously.)

- RAMWEN1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RWN1KEY[7:0]								RAM WE15	RAM WE14	RAM WE13	RAM WE12	RAM WE11	RAM WE10	RAM WE9	RAM WE8
Initial value:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
R/W:	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: To avoid erroneous rewriting, the way of writing data to this register is different from that to other general registers. For details, see section 30.2.7, Notes on Register Access.

Bit	Bit Name	Initial Value	R/W	Descriptions
15 to 8	RWN1KEY [7:0]	H'00	R/(W)	These bits enable or disable write to the RAMWE bit. H'68: Enable write to bits RAMWE15 to RAMWE8. The write data is not retained and these bits are always read as H'00. Other than H'68: Disable write to bits RAMWE15 to RAMWE8.
7	RAMWE15	1	R/W	RAM Write Enable 15 Enables or disables write to page 15 in the on-chip RAM. 0: Disables write to page 15 in the on-chip RAM 1: Enables write to page 15 in the on-chip RAM. [Clearing condition] • Writing 0 (H'68 is written to the upper byte simultaneously.) [Setting conditions] • Reset or standby • Writing 1 (H'68 is written to the upper byte simultaneously.)

Bit	Bit Name	Initial Value	R/W	Descriptions
6	RAMWE14	1	R/W	RAM Write Enable 14 Enables or disables write to page 14 in the on-chip RAM. 0: Disables write to page 14 in the on-chip RAM 1: Enables write to page 14 in the on-chip RAM [Clearing condition] • Writing 0 (H'68 is written to the upper byte simultaneously.) [Setting conditions] • Reset or standby • Writing 1 (H'68 is written to the upper byte simultaneously.)
5	RAMWE13	1	R/W	RAM Write Enable 13 Enables or disables write to page 13 in the on-chip RAM. 0: Disables write to page 13 in the on-chip RAM 1: Enables write to page 13 in the on-chip RAM [Clearing condition] • Writing 0 (H'68 is written to the upper byte simultaneously.) [Setting conditions] • Reset or standby • Writing 1 (H'68 is written to the upper byte simultaneously.)

Bit	Bit Name	Initial Value	R/W	Descriptions
4	RAMWE12	1	R/W	RAM Write Enable 12 Enables or disables write to page 12 in the on-chip RAM. 0: Disables write to page 12 in the on-chip RAM 1: Enables write to page 12 in the on-chip RAM [Clearing condition] • Writing 0 (H'68 is written to the upper byte simultaneously.) [Setting conditions] • Reset or standby • Writing 1 (H'68 is written to the upper byte simultaneously.)
3	RAMWE11	1	R/W	RAM Write Enable 11 Enables or disables condition for write to page 11 in the on-chip RAM. 0: Disables write to page 11 in the on-chip RAM 1: Enables write to page 11 in the on-chip RAM [Clearing condition] • Writing 0 (H'68 is written to the upper byte simultaneously.) [Setting conditions] • Reset or standby • Writing 1 (H'68 is written to the upper byte simultaneously.)

Bit	Bit Name	Initial Value	R/W	Descriptions
2	RAMWE10	1	R/W	RAM Write Enable 10 Enables or disables write to page 10 in the on-chip RAM. 0: Disables write to page 10 in the on-chip RAM 1: Enables write to page 10 in the on-chip RAM [Clearing condition] • Writing 0 (H'68 is written to the upper byte simultaneously.) [Setting conditions] • Reset or standby • Writing 1 (H'68 is written to the upper byte simultaneously.)
1	RAMWE9	1	R/W	RAM Write Enable 9 Enables or disables write to page 9 in the on-chip RAM. 0: Disables write to page 9 in the on-chip RAM 1: Enables write to page 9 in the on-chip RAM [Clearing condition] • Writing 0 (H'68 is written to the upper byte simultaneously.) [Setting conditions] • Reset or standby • Writing 1 (H'68 is written to the upper byte simultaneously.)

Bit	Bit Name	Initial Value	R/W	Descriptions
0	RAMWE8	1	R/W	RAM Write Enable 8 Enables or disables write to page 8 in the on-chip RAM. 0: Disables write to page 8 in the on-chip RAM 1: Enables write to page 8 in the on-chip RAM [Clearing condition] • Writing 0 (H'68 is written to the upper byte simultaneously.) [Setting conditions] • Reset or standby • Writing 1 (H'68 is written to the upper byte simultaneously.)

30.2.3 RAM ECC Enable Control Register (RAMECC)

The RAMECC enable control register (RAMECC) enables or disables the ECC correction function. RAMECC is initialized by a reset or in the standby state.

RAMECC can be written to in words, and can be read in bytes or words. To rewrite to the RAMECC bits, word size data with upper byte as H'76 and lower byte as write data must be written.

When the upper byte (bits 15 to 8) of RAMECC is read, the read value is always H'00.

Do not place an instruction to access to the on-chip RAM immediately after an instruction to write to RAMECC; otherwise correct access to the on-chip RAM cannot be guaranteed.

When rewriting to the RECCA bit, execute an instruction to read from RAMECC together with five or more NOP instructions immediately after an instruction to write to RAMECC.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	REKEY[7:0]								-	-	-	-	-	-	-	RECCA
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R	R	R	R	R	R	R	R/W

Note: To avoid erroneous rewriting, the way of writing data to this register is different from that to other general registers. For details, see section 30.2.7, Notes on Register Access.

Bit	Bit Name	Initial Value	R/W	Descriptions
15 to 8	REKEY[7:0]	H'00	R/(W)	These bits enable or disable write to the RECCA bit. H'76: Enable write to the RECCA bit. The write data is not retained and these bits are always read as H'00. Other than H'76: Disable write to the RECCA bit.
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Descriptions
0	RECCA	0	R/W	Enables or disables the ECC correction. 0: Enables ECC correction [Clearing conditions] • Reset or standby • Writing 0 (H'76 is written to the upper byte simultaneously.) 1: Disables ECC correction [Setting condition] • Writing 1 (H'76 is written to the upper byte simultaneously.)

30.2.4 RAM Error Status Register (RAMERR)

The RAM error status register (RAMERR) monitors RAM error occurrence.

RAMERR is initialized by a reset or upon entering a standby state. RAMERR can be read or written to in bytes.

When the ECC error correction function is enabled, a 1-bit error correction in page 7 to page 0 sets the RCRTC0 bit and one in page 15 to page 8 sets the RCRTC1 bit during RAM read operation. 2-bit error detection in page 7 to page 0 sets the RDTCT0 bit and that in page 15 to page 8 sets the RDTCT1 bit. When the ECC error correction function is disabled, a parity error in page 7 to page 0 sets the RPARI0 bit and one in page 15 to page 8 sets the RPARI1 bit during an on-chip RAM read operation.

If the ECC error correction function is disabled via RAMECC after the RDTCT0 RDTCT1, RCRTC0, and RCRCT1 bits are set, these bits remain set. If the ECC error correction function is enabled via RAMECC after the RPARI0 and RPARI1 bits are set, these bits remain set.

To clear the status bits that have been set, be sure to write 0 to them after reading 1 from them. Execute an instruction to read RAMERR together with five or more NOP instructions after writing 0.

Bits 7 and 6 in RAMERR are always read as 0.

Bit:	7	6	5	4	3	2	1	0
	-	-	RP ARI1	RP ARI0	RD TCT1	RC RCT1	RD TCT0	RC RCT0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Descriptions
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	RPARI1	0	R/W	RAM parity error monitor bit 1 Monitors whether or not a parity error occurs in page 15 to page 8 when the ECC error correction is disabled. 0: No parity error has occurred. 1: A parity error has occurred. [Clearing conditions] • Reset or standby • Writing 0 after reading 1 [Setting condition] A parity error has occurred
4	RPARI0	0	R/W	RAM parity error monitor bit 0 Monitors whether or not a parity error occurs in page 7 to page 0 when the ECC error correction is disabled. 0: No parity error has occurred. 1: A parity error has occurred. [Clearing conditions] • Reset or standby • Writing 0 after reading 1 [Setting condition] • A parity error has occurred

Bit	Bit Name	Initial Value	R/W	Descriptions
3	RDTCT1	0	R	RAM 2-bit error detection monitor bit 1 Monitors whether or not 2-bit error detection occurs in page 15 to page 8 when the ECC error correction is enabled. 0: No 2-bit error detection has occurred. 1: 2-bit error detection has occurred. [Clearing conditions] • Reset or standby • Writing 0 after reading 1 [Setting condition] • 2-bit error detection has occurred
2	RCRCT1	0	R/W	RAM 1-bit error detection monitor bit 1 Monitors whether or not a 1-bit error correction occurs in page 15 to page 8 when the ECC error correction is enabled. 0: No 1-bit error correction has occurred. 1: A 1-bit error correction has occurred. [Clearing conditions] • Reset or standby • Writing 0 after reading 1 [Setting condition] • A 1-bit error correction has occurred
1	RDTCT0	0	R/W	RAM 2-bit error detection monitor bit 0 Monitors whether or not 2-bit error detection occurs in page 7 to page 0 when the ECC error correction is enabled. 0: No 2-bit error detection has occurred. 1: 2-bit error detection has occurred. [Clearing conditions] • Reset or standby • Writing 0 after reading 1 [Setting condition] • 2-bit error detection has occurred

Bit	Bit Name	Initial Value	R/W	Descriptions
0	RCRCT0	0	R/W	RAM 1-bit error correction monitor bit 0 Monitors whether or not a 1-bit error correction occurs in page 7 to page 0 when the ECC error correction is enabled. 0: No 1-bit error correction has occurred. 1: A 1-bit error correction has occurred. [Clearing conditions] • Reset or standby • Writing 0 after reading 1 [Setting condition] • A 1-bit error correction has occurred

30.2.5 RAM Error Interrupt Control Register (RAMINT)

The RAM error interrupt control register (RAMINT) enables or disables the RAM error interrupt.

If the RECIE0 and RECIE1 bits of RAMINT are set enabled while the ECC error correction function is set enabled via RAMECC, an interrupt is generated upon the correction of a 1-bit error or detection of a 2-bit error. Also, if the REDIE0 and REDIE1 bits of RAMINT are set enabled, an interrupt is generated upon detection of a 2-bit error. Table 30.3 shows the conditions required for interrupt generation when the ECC error correction is set enabled. If the RPEIE0 and RPEIE1 bits of RAMINT are set enabled while the ECC error correction function is set enabled via RAMECC, an interrupt is generated upon occurrence of a parity error.

RECIE0, REDIE0, and RPEIE0 correspond to the user RAM area of page 7 to page 0. Also, RECIE1, REDIE1, and RPEIE1 correspond to that of page 15 to page 0.

RAMINT is initialized by a reset or upon entering the standby state. RAMINT can only be read or written to in bytes. When rewriting RAMINT, execute an instruction to read from RAMINT together five or more NOP instructions immediately after an instruction to write to RAMINT.

Bit:	7	6	5	4	3	2	1	0
	-	-	RP EIE1	RP EIE0	RE DIE1	RE CIE1	RE DIE0	RE CIE0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Descriptions
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	RPEIE1	0	R/W	RAM parity error interrupt bit 1 Enables/disables an interrupt upon occurrence of a parity error in page 15 to page 8 when the ECC error correction is disabled. 0: Disables an interrupt upon occurrence of a parity error. 1: Enables an interrupt upon occurrence of a parity error.

Bit	Bit Name	Initial Value	R/W	Descriptions
4	RPEIE0	0	R/W	RAM parity error interrupt bit 0 Enables/disables an interrupt upon occurrence of a parity error in page 7 to page 0 when the ECC error correction is disabled. 0: Disables an interrupt upon occurrence of a parity error. 1: Enables an interrupt upon occurrence of a parity error.
3	REDIE1	0	R/W	RAM 2-bit error detection interrupt bit 1 Enables/disables an interrupt upon detection of a 2-bit error in page 15 to page 8 when the ECC error correction is enabled. 0: Disables an interrupt upon detection of a 2-bit error. 1: Enables an interrupt upon detection of a 2-bit error.
2	RECIE1	0	R/W	RAM 1-bit error correction interrupt bit 1 Enables/disables an interrupt upon occurrence of a 1-bit error correction in page 15 to page 8 when the ECC error correction is enabled. 0: Disables an interrupt upon occurrence of a 1-bit error correction. 1: Enables an interrupt upon occurrence of a 1-bit error correction.

Bit	Bit Name	Initial Value	R/W	Descriptions
1	REDIE0	0	R/W	RAM 2-bit error detection interrupt bit 0 Enables/disables an interrupt upon detection of a 2-bit error in page 7 to page 0 when the ECC error correction is enabled. 0: Disables an interrupt upon detection of a 2-bit error. 1: Enables an interrupt upon detection of a 2-bit error.
0	RECIE0	0	R/W	RAM 1-bit error correction interrupt bit 0 Enables/disables an interrupt upon occurrence of a 1-bit error correction in page 7 to page 0 when the ECC error correction is enabled. 0: Disables an interrupt upon occurrence of a 1-bit error correction. 1: Enables an interrupt upon occurrence of a 1-bit error correction.

Table 30.3 Conditions for Interrupt Occurrence When ECC Error Correction is Enabled

REDIE Bit	RECIE Bit	RAM Errors to Trigger an Interrupt
0	0	None
0	1	Upon occurrence of a 1-bit error correction or 2-bit error detection
1	0	Upon occurrence of 2-bit error correction
1	1	Upon occurrence of a 1-bit error correction or 2-bit error detection

30.2.6 RAM Access Cycle Set Register (RAMACYC)

The RAM access cycle set register (RAMACYC) sets the RAM read or write cycle. The cycle setting range depends on the PLL clock multiplication ratio and ECC enable/disable. For details, refer to the recommended setting range shown in table 30.4.

The WRCYC and RDCYC bits are initialized by a reset or in the standby state.

To write the RAMACYC bits, word size data with upper byte as H'78 and lower byte as write data must be written. RAMACYC can be written to in words, and can be read in bytes or words.

When the upper byte (bits 15 to 8) of RAMACYC is read, the read value is always H'00.

Do not write to RAMACYC while RAM is being accessed. Be sure to set all the RAME bits in the RAM enable control register to 0 to disable access to the RAM before rewriting RAMACYC. Also, do not place an instruction to access to the on-chip RAM immediately after an instruction to write to RAMACYC; otherwise normal access is not guaranteed.

When rewriting the WRCYC1, WRCYC0 and RDCYC bits, execute an instruction to read from RAMACYC together with five or more NOP instructions immediately after an instruction to write to RAMACYC.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RAKEY[7:0]								-	-	WRCYC[1:0]		-	-	-	RDCYC
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R	R	R/W	R/W	R	R	R	R/W

Note: To avoid erroneous rewriting, the way of writing data to this register is different from that to other general registers. For details, see section 30.2.7, Notes on Register Access.

Bit	Bit Name	Initial Value	R/W	Descriptions
15 to 8	RAKEY[7:0]	H'00	R/(W)	These bits enable or disable write to bits WRCYC[1], WRCYC[0], and RDCYC. H'78: Enable write to bits WRCYC[1], WRCYC[0], and RDCYC. The write data is not retained and these bits are always read as H'00. Other than H'78: Disable write to bits WRCYC[1], WRCYC[0], and RDCYC.
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Descriptions
5, 4	WRCYC[1:0]	00	R/W	These bits set the RAM write cycle. 00: Set write cycle to 4 cycles 01: Set write cycle to 3 cycles 10: Set write cycle to 2 cycles 11: Setting prohibited [Clearing conditions] • Reset or standby • Writing B'00 (H'78 is written to the upper byte simultaneously.) [Setting condition] • H'78 is written to the upper byte simultaneously. Note: Do not set for 2 cycles unless ×4 multiplication has been set for the PLL.
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	RDCYC	0	R/W	Sets the RAM read cycle. 0: Set read cycle to 2 cycles [Clearing conditions] • Reset or standby • Writing 0 (H'78 is written to the upper byte simultaneously.) 1: Set read cycle to 1 cycle [Setting condition] • Writing 1 (H'78 is written to the upper byte simultaneously.) Note: Do not set for 1 cycle unless ×4 multiplication has been set for the PLL.

Accesses through the I bus are ruled by the peripheral clock (Pϕ); this is not the case with M or F buses, which are ruled by the internal clock (ϕ). Therefore the number of cycles required for an I-bus access is determined by combination of the set PLL and peripheral clock multiplication ratios as well as the set value in this register.

It takes one cycle to receive a command or address during I bus read access on condition that the PLL multiplication ratio is ×4 or ×6 or on condition that the PLL multiplication ratio is ×8 or ×10 and the peripheral clock multiplication ratio is ×2. It also takes one cycle during I bus write access on condition that the PLL multiplication ratio is ×4 or ×6 and the peripheral clock multiplication ratio is ×2. This is why the entire process, receiving a command and address, needs two cycles on such conditions.

Table 30.4 Recommended Settings for Read and Write Cycles

PLL Multiplication Ratio	Recommended Settings		F Bus/M Bus Access (φ)		Peripheral Clock Multiplication Ratio	I Bus Access (Pφ)	
	RDCYC	WRCYC[1:0]	Read [Cycle]	Write [Cycle]		Read [Cycle]	Write [Cycle]
×4	B'1	B'10	1	2	×1	2	1
					×2	2	2
×6	B'0	B'01	2	3	×1	2	1
					×2	2	2
×8			2	3	×1	1	1
					×2	2	1
×10			2	3	×1	1	1
					×2	2	1

30.2.7 Notes on Register Access

The way of writing data to the RAM enable control registers 0 and 1 (RAMEN0 and RAMEN1), RAM write enable control registers 0 and 1 (RAMWEN), RAM ECC enable control register (RAMECC), and RAM access cycle set register (RAMACYC) is different from that to other general registers. This is because these registers are not to be easily rewritten.

To write these registers, use the following ways. In addition, note that RAMEN0, RAMEN1, RAMWEN0, RAMWEN1, RAMECC, and RAMACYC must be written in words. These registers cannot be written in byte or longword instructions. As shown in figure 30.3, key data should be written in the upper byte.

- To write data to RAMEN0, transfer data with upper byte as H'96 and lower byte as write data.
- To write data to RAMEN1, transfer data with upper byte as H'95 and lower byte as write data.
- To write data to RAMWEN0, transfer data with upper byte as H'69 and lower byte as write data.
- To write data to RAMWEN1, transfer data with upper byte as H'68 and lower byte as write data.
- To write data to RAMECC, transfer data with upper byte as H'76 and lower byte as write data.
- To write data to RAMACYC, transfer data with upper byte as H'78 and lower byte as write data.

When the upper bytes (bits 15 to 8) of RAME0, RAME1, RAMWEN0, RAMWEN1, RAMECC, and RAMACYC are read, the read value is always H'00.

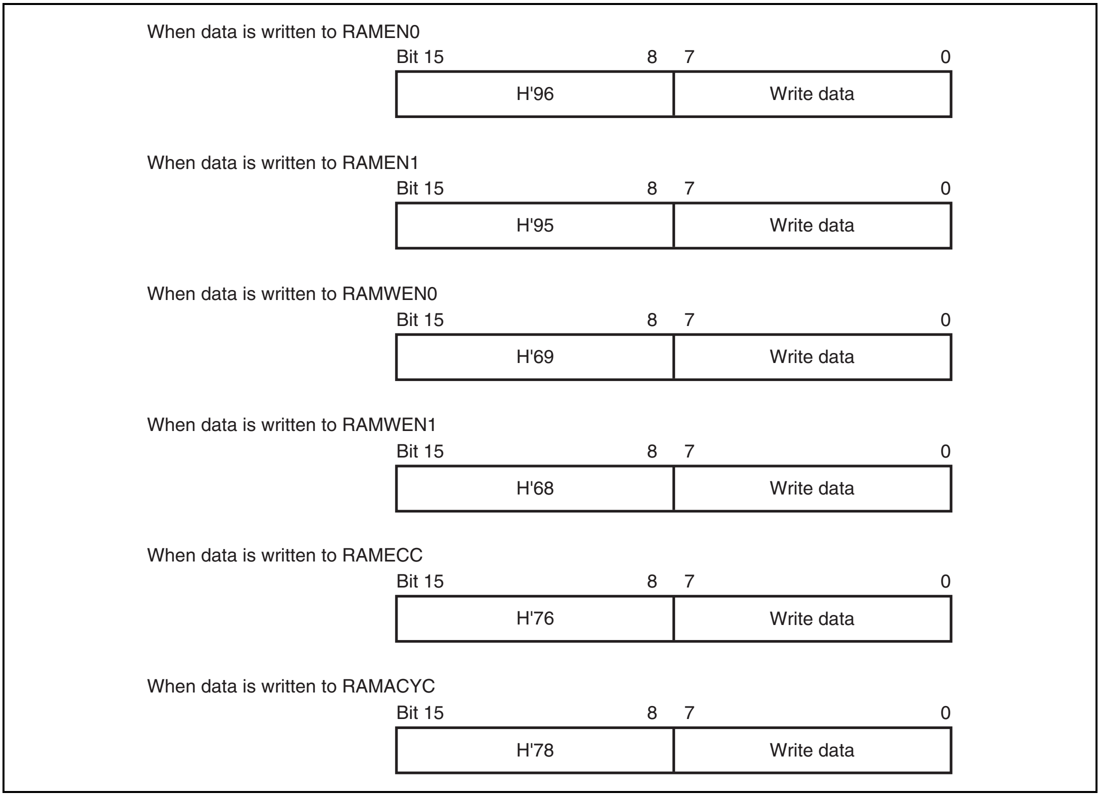

When data is written to RAMEN0
Bit 15
8
7
0
H'96
Write data
When data is written to RAMEN1
Bit 15
8
7
0
H'95
Write data
When data is written to RAMWEN0
Bit 15
8
7
0
H'69
Write data
When data is written to RAMWEN1
Bit 15
8
7
0
H'68
Write data
When data is written to RAMECC
Bit 15
8
7
0
H'76
Write data
When data is written to RAMACYC
Bit 15
8
7
0
H'78
Write data

Figure 30.3 Writing Data to RAMEN0, RAMEN1, RAMWEN0, RAMWEN1, RAMECC, and RAMACYC

30.3 On-Chip RAM Operations

Access to the on-chip RAM is controlled by the RAM enable control registers 0 and 1 (RAMEN0 and RAMEN1) and RAM write enable control registers 0 and 1 (RAMWEN0 and RAMWEN1).

Accessing each area of the on-chip RAM is enabled or disabled by the RAME15 to RAME0 bits in the RAM enable control registers 0 and 1 (RAMEN0 and RAMEN1). When the RAME15 to RAME0 bits of RAMEN0 and RAMEN1 are cleared to 0, the on-chip RAM cannot be accessed. In this case, values read from the on-chip RAM are undefined and the on-chip RAM cannot be modified.

Writing to each area of the on-chip RAM is enabled or disabled by the RAMWE15 to RAMWE0 bits in the RAM write enable control registers 0 and 1 (RAMWEN0 and RAMWEN1).

The ECC error correction function can be set enabled or disabled by register settings. The function is initially set enabled.

When the ECC error correction function is set enabled, 1-bit error correction and 2-bit error detection can be performed. Upon correction of a 1-bit error and/or detection of a 2-bit error, flags (in the RAM error status register) are set to indicate their occurrence. When the ECC error correction is set disabled, a flag (in the RAM error status register) is set upon occurrence of a parity error. Whenever the RAM error status register is set, a RAM error (RAME) interrupt can be generated. The interrupt generation can be set enabled or disabled via the RAM error interrupt control register (RAMINT).

30.4 RAM Data Retention

30.4.1 Data Retention at Reset

If a low level signal is input on the $\overline{\text{RES}}$ pin from an external device while this LSI is in operation, this LSI enters the power-on reset state. In this case, if the on-chip RAM is accessed, data in the RAM address being accessed may be destroyed because the bus cycle cannot be completed normally.

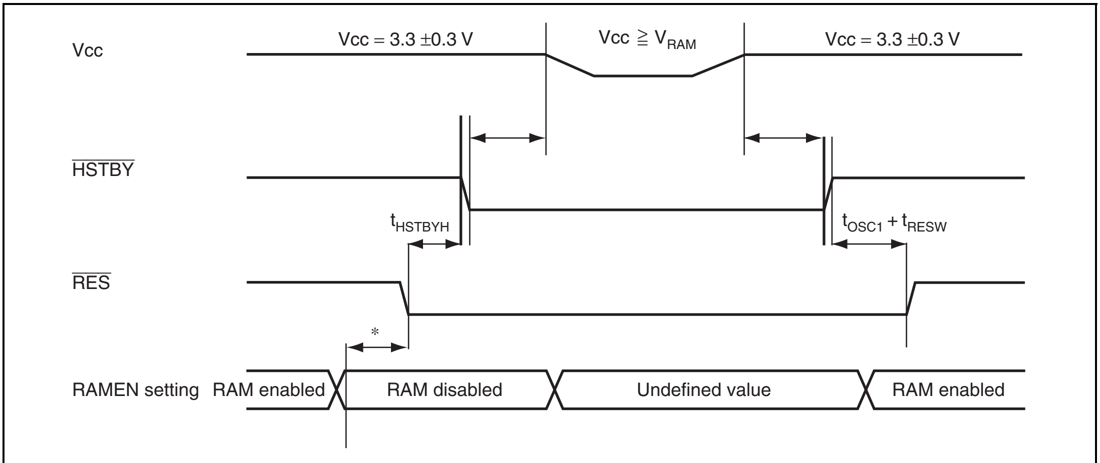
Since it is difficult to input reset signals from the external devices while avoiding accesses to the on-chip RAM. Accordingly, to retain all data items in the on-chip RAM at reset, invalidate the RAM by the RAM enable control register (RAMEN).

However, data in the on-chip RAM is not retained when the operation enters boot mode, user boot mode, or writer mode from the power-on reset state since the RAM is cannot be disabled as occupied by programs embedded in the LSI.

30.4.2 Data Retention at Hardware Standby

This LSI turns off the internal power supply except for pages 0 and 1 in the on-chip RAM if the hardware standby state is entered. Accordingly, data in the on-chip RAM other than that in pages 0 and 1 cannot be retained. To retain data in pages 0 and 1 during hardware standby state, hardware standby state must be entered according to the sequence shown in figure 30.4.

During hardware standby state, data in pages 0 and 1 can be retained while the specified voltage (V_{RAM}) is supplied to the Vcc power supply.



Note: * The RAMEN setting must be invalidated before a low-level signal is input to the RES pin.

Figure 30.4 RAM Data Retention at Hardware Standby

30.5 Notes on Usage

30.5.1 Page Conflict

If the same page is accessed by the different buses simultaneously, a page conflict occurs. Each of those accesses is handled in such priority scheme as: I bus (highest), M bus (middle), F bus (lowest).

In this case, each access is completed normally but this conflict degrades the memory access efficiency. To avoid this conflict, it is recommended to take preventative measures by software. For example, accessing different memory or different pages using different buses can avoid page conflict.

30.5.2 State After Turning on Power

After turning on the power, all data items in the on-chip RAM including ECC correction data and parity are undefined. Accordingly, correspondence among RAM data, error correction data, and parity may not be correct.

To enable correct (initialize) correspondence among RAM data, error correction data, and parity after power-on, write data to all on-chip RAM areas. If RAM is read without initialization, a RAM error may occur. Note that no RAM error occurs in RAM writes.

30.5.3 Write Operation When Writing RAM is Disabled

When a write operation is performed to a page with RAM writing disabled in initialized RAM and the address of the page has encountered an ECC error, an ECC error flag is set due to wrongly detected ECC error. Also interrupts are generated if interrupts are enabled. In that case, however, the data are not destroyed since the write operation is not performed.

Section 31 Power-Down Modes

To reduce device power consumption, this device incorporates two types of power down modes: hardware standby mode and sleep mode, and a module standby function for halting the operation of certain modules. A proper mode or function should be selected according to the application.

31.1 Features

31.1.1 Power-Down Modes

This LSI has the following power-down modes and function:

- Hardware standby mode

  This LSI enters hardware standby mode by inputting specific levels to the $\overline{\text{RES}}$ and $\overline{\text{HSTBY}}$ pins. In this mode, all the functions of the LSI halt and power supply to most part of the internal LSI is halted. The LSI returns from hardware standby mode by a power-on reset.
- Sleep mode

  The LSI enters sleep mode by the CPU instruction. In this mode, the on-chip peripheral modules other than the CPU operate. The LSI returns from sleep mode by a power-on reset, an interrupt, or a DMA address error.
- Module standby function

  The operation of on-chip peripheral modules that can enter the module standby state (FPU, UBC, DMAC, AUD-II, JTAG Interface, and FlexRay) is halted by stopping clock supply. The clock supply to the target module can be controlled by the corresponding bit of STBCR.

Table 31.1 shows states of the CPU and peripheral modules in each mode. Table 31.2 shows the transition conditions for entering each mode from the program execution state, as well as the procedures for revoking each mode.

Table 31.1 States of Power-Down Modes

	State*				
Power-Down Mode	CPG	CPU	CPU Register	On-Chip RAM	On-Chip Peripheral Modules
Hardware standby mode	Halts (Power supply halted)	Halts (Power supply halted)	Halts (Power supply halted)	A part of RAM area (32 Kbytes) is retained. (Power supply halted outside the retained area.)	Halts (Power supply halted)
Sleep mode	Runs	Halts	Held	Runs	Runs
Module standby function	Runs	Runs	Runs	Runs	Specified module halts

Note: * The pin state is retained or set to high impedance. For details, see appendix A, Pin States.

Table 31.2 Transition Conditions for Entering Power-Down Mode and Revoking Procedures

Power-Down Mode	Transition Conditions	Revoking Procedure
Hardware standby mode	• Driving the $\overline{\text{HSTBY}}$ pin low while the $\overline{\text{RES}}$ pin is low	• Power-on reset after inputting a high level signal on the $\overline{\text{HSTBY}}$ pin
Sleep mode	• Executing SLEEP instruction	• Power-on reset • Interrupt • DMA address error
Module standby function	• Setting the MSTP bits of STBCR to 1	• Clearing the MSTP bit to 0 • Power-on reset

31.2 Input/Output Pins

Table 31.3 shows the pin configuration related to power-down modes.

Table 31.3 Pin Configuration

Pin Name	Symbol	I/O	Function
Power-on reset	$\overline{\text{RES}}$	Input	Inputting a low level signal on this pin causes a transition to power-on reset processing.
Hardware standby	$\overline{\text{HSTBY}}$	Input	Inputting a low level signal on this pin while the $\overline{\text{RES}}$ pin is low causes a transition to hardware standby mode.

31.3 Register Descriptions

The following register is used in power-down modes.

Table 31.4 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Standby control register	STBCR	R/W	H'0020	H'FFFE0400	8, 16

31.3.1 Standby Control Register (STBCR)

STBCR is a 16-bit readable/writable register that controls the operation of each module in power-down modes. This register is initialized to H'0020 by a power-on reset or in hardware standby mode.

STBCR must be written to in word units. When rewriting the values of the MSTP5 to MSTP0 bits, write H'3C to the STBCRKEY bits simultaneously. If data other than H'3C is written to the STBCRKEY bits or data is written in byte units, the operation is ignored.

Both word and byte accesses are available to read STBCR. However, the STBCRKEY bits are always read as H'00 because the values written to the STBCRKEY bits are not retained.

Note: To prevent erroneous rewriting, the way of writing data to this register is different from that for other general registers. For details, refer to section 31.3.2, Note on Accessing STBCR.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	STBCRKEY								-	-	MSTP5	MSTP4	MSTP3	MSTP2	MSTP1	MSTP0
Initial value:	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Note: * The write value is not retained.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	STBCRKEY	H'00	R/W	STBCR Write Key Code These bits enable or disable modifying bits MSTP5 to MSTP0. These bits are always read as H'00 because the values written to these bits are not retained. H'3C: Enable modifying bits MSTP5 to MSTP0 Other than H'3C: Disable modifying bits MSTP5 to MSTP0
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	MSTP5	1	R/W	Module Stop 5 When this bit is set to 1, the clock supply to the FlexRay is halted. When this bit is cleared to 0, the clock supply to the FlexRay is restarted. Note that the registers in the FlexRay are not initialized even when the clock supply to the FlexRay is halted. 0: FlexRay runs 1: Clock supply to FlexRay is halted
4	MSTP4	0	R/W	Module Stop 4 Setting this bit to 1 puts the JTAG TAP controller into a reset state. Clearing this bit to 0 releases the TAP controller from its reset state and brings back into operation in accordance with input on the JTAG pins. 0: JTAG TAP module runs 1: JTAG TAP module is in a reset state

Bit	Bit Name	Initial Value	R/W	Description
3	MSTP3	0	R/W	Module Stop 3 When this bit is set to 1, the clock supply to the DMAC is halted. When this bit is cleared to 0, the clock supply to the DMAC is restarted. Note that the registers in the DMAC are not initialized even when the clock supply to the DMAC is halted. 0: DMAC runs 1: Clock supply to DMAC is halted
2	MSTP2	0	R/W	Module Stop 2 When the MSTP2 bit is set to 1, the clock supply to the AUD-II is halted. When this bit is cleared to 0, the clock supply to the AUD-II is restarted. The internal state (including registers) of the AUD-II is initialized when the clock supply to the AUD-II is halted. However, if the trace function is enabled (the EN bit of AUCSR is set to 1) while the AUD-II operates in trace mode, the AUD-II does not enter the module standby state. Even if the trace function is disabled, the AUD-II does not enter the module standby state as long as there remains acquired trace data in the internal FIFO. The AUD-II enters the module standby state after all remaining data has been output from the FIFO. 0: AUD-II runs 1: Clock supply to AUD-II is halted
1	MSTP1	0	R/W	Module Stop 1 When this bit is set to 1, the clock supply to the FPU is halted. Once this bit is set to 1, it is impossible to write 0 to this bit; that is to say, once the clock supply to the FPU is halted by setting this bit to 1, the clock supply to the FPU cannot be restarted by clearing this bit to 0. To restart the clock supply to the FPU, reset the LSI by a power-on reset. 0: FPU runs 1: Clock supply to FPU is halted

Bit	Bit Name	Initial Value	R/W	Description
0	MSTP0	0	R/W	Module Stop 0 When this bit is set to 1, the clock supply to the UBC is halted. When this bit is cleared to 0, the clock supply to the UBC is restarted. The registers in the UBC are not initialized even when the clock supply to the UBC is halted. 0: UBC runs 1: Clock supply to UBC is halted

31.3.2 Note on Accessing STBCR

The writing procedure for STBCR is different from that for other general registers to prevent it being modified easily. This register should be written to or read in the following procedure.

When writing to STBCR, be sure to use a word transfer instruction. Data cannot be written to by a byte transfer instruction. As shown in figure 31.1, transfer the data with the upper byte as H'3C and the lower byte as the write data. This transfer procedure writes the lower byte data to STBCR.

The reading procedure is similar to that for general registers. STBCR is allocated at address H'FFFE0400. Both byte transfer instructions and word transfer instructions are available.

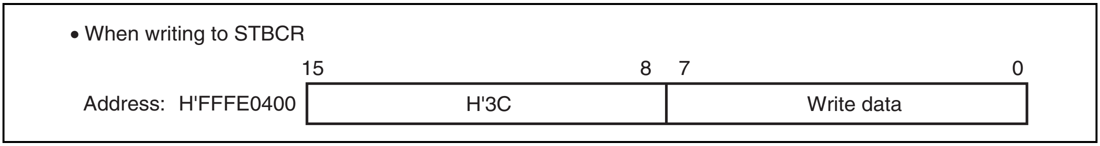


Figure 31.1 Writing to STBCR

31.4 Operation

31.4.1 Hardware Standby Mode

(1) Transition to Hardware Standby Mode

This LSI enters hardware standby mode by setting the $\overline{\text{HSTBY}}$ pin low after setting the $\overline{\text{RES}}$ pin low. Set the mode pins as described in section 3, Operating Modes; otherwise, the correct operation is not guaranteed.

In hardware standby mode, all the functions of the LSI halt and the internal power supply to the area other than pages 0 and 1 of the on-chip RAM is halted, which reduces the LSI power consumption greatly. This LSI enters hardware standby mode asynchronously with and regardless of its current state because the transition is caused by an input on the external pin. Therefore, the LSI state before entering hardware standby mode is not retained except pages 0 and 1 of the on-chip RAM.

The data stored in pages 0 and 1 of the on-chip RAM can be retained as long as the specific voltage is applied. To retain data stored in the on-chip RAM, clear the RAME0 bit (for page 0 area) and the RAME1 bit (for page 1 area) in the RAM enable control register (RAMEN) before setting the $\overline{\text{HSTBY}}$ pin low. For details on RAMEN, see section 30, RAM.

Hold the $\overline{\text{HSTBY}}$ pin low in hardware standby mode.

For the register states in hardware standby mode, see section 33, List of Registers. For the pin states in hardware standby mode, refer to appendix A, Pin States.

(2) Revoking Hardware Standby Mode

Hardware standby mode can be revoked only with the $\overline{\text{HSTBY}}$ and $\overline{\text{RES}}$ pins.

Setting the $\overline{\text{HSTBY}}$ pin high while the $\overline{\text{RES}}$ pin is low starts the clock oscillation. Be sure to hold the $\overline{\text{RES}}$ pin low until the clock oscillation is stabilized. After the clock oscillation has been stabilized, setting the $\overline{\text{RES}}$ pin high starts the power-on reset exception handling by the CPU.

(3) Timing in Hardware Standby Mode

Figure 31.2 shows an example of the timing of each pin in hardware standby mode.

This LSI enters hardware standby mode by setting the $\overline{\text{HSTBY}}$ pin low after setting the $\overline{\text{RES}}$ pin low. To revoke the mode, set the $\overline{\text{HSTBY}}$ pin high, and after the clock has been stabilized set the $\overline{\text{RES}}$ pin high.

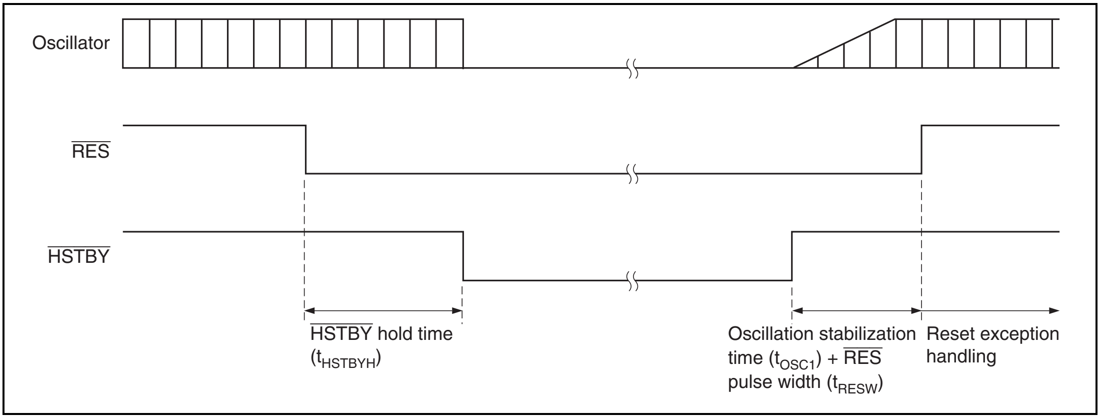

Oscillator
RES
HSTBY
HSTBY hold time (t_{HSTBYH})
Oscillation stabilization time (t_{OSC1}) + RES pulse width (t_{RESW})
Reset exception handling

Figure 31.2 Timing in Hardware Standby Mode

31.4.2 Sleep Mode

(1) Transition to Sleep Mode

Execution of the SLEEP instruction by the CPU causes a transition from the program execution state to sleep mode. Although the CPU halts immediately after executing the SLEEP instruction, the contents of its internal registers remain unchanged. The on-chip peripheral modules continue to run in sleep mode. Clock pulses continue to be output even in sleep mode if output from the CK pin is enabled by the CK control register (CKCR). For details on CKCR, refer to section 24, I/O Ports. For the state of each register in sleep mode, refer to section 33, List of Registers.

(2) Revoking Sleep Mode

Sleep mode is revoked by an interrupt (NMI, IRQ, and on-chip peripheral module), DMA address error, or power-on reset.

- Revoking with an interrupt

  When an NMI, IRQ, or on-chip peripheral module interrupt occurs, sleep mode is revoked and interrupt exception handling is executed. When the priority level of the generated interrupt is equal to or lower than the interrupt mask level that is set in the status register (SR) of the CPU, or the interrupt by the on-chip peripheral module is disabled on the module side, the interrupt request is not accepted and sleep mode is not revoked.

- Revoking with a DMA address error

  When a DMA address error occurs, sleep mode is revoked and DMA address error exception handling is executed.
- Revoking with a reset

  When a low signal is input on the $\overline{\text{RES}}$ pin or an internal reset by the WDT occurs, this LSI enters the power-on reset state and sleep mode is revoked.

31.4.3 Module Standby Function

(1) Transition to Module Standby Function

Setting the MSTP bits in the standby control register to 1 halts the supply of clocks to the corresponding on-chip peripheral modules. This function can be used to reduce the power consumption in normal mode and sleep mode. Disable a module before placing it in the module standby mode. In addition, do not access the module's registers while it is in the module standby state.

The external pin states of the module in the module standby state vary. For details, see appendix A, Pin States.

The contents of registers in the UBC and the DMAC are retained even in the module standby state.

The JTAG TAP controller is put into a reset state (Test-Logic-Reset state) by the module standby function.

The registers in the FlexRay are not initialized even in the module standby state.

The internal state (including registers) of the AUD-II is initialized by the module standby function. However, the AUD-II does not enter the module standby state if the trace function is enabled (the EN bit in AUCSR is set to 1) while the AUD-II is operated in trace mode. Even if the trace function is disabled, the AUD-II does not enter in the module standby state as long as there remains the acquired trace data in the internal FIFO. The AUD-II switches to the module standby state after all remaining data has been output from the FIFO.

The FPU cannot restart operation once it has entered the module standby state. To restart the module, reset this LSI by a power-on reset.

(2) Revoking Module Standby Function

The module standby function can be revoked by clearing the MSTP bits to 0, or by a power-on reset. When taking a module out of the module standby state by clearing the corresponding MSTP bit to 0, read the MSTP bit to confirm that it has been cleared to 0.

Section 32 Reliability

32.1 Reliability

A failure rate curve represents an index of the reliability of a semiconductor device. The failure rate curve traces a bathtub shape over the course of time, as is shown in figure 32.1. The curve is divided into three periods according to the type of failure phenomena: an initial failure period, a random failure period (functional lifetime), and a wear-out failure period. Initial failures, which occur during the initial failure period, are caused by contamination with foreign matter and localized chemical pollution; these can be eliminated by screening. Wear-out failures in the final period are caused by the deterioration of materials that make up semiconductor devices during long periods of usage. Random failures, which occur during the random failure period, are thought to occur in cases where a device with a minor failure is not removed by screening, and so is shipped, and then fails during the customer's production process or in the field, and in cases where a failure which should normally not have occurred until the wear-out period occurs earlier because of variations in production. Therefore, the reliability of semiconductor device is secured by appropriate screening to reduce the presence of initial failures and high reliability design to prevent the occurrence of wear-out failures. The reliability of a product is confirmed by producing a large quantity of prototypes for checking of the initial failure rate and executing accelerated life testing to identify the wear-out failure time in a realistic environment.

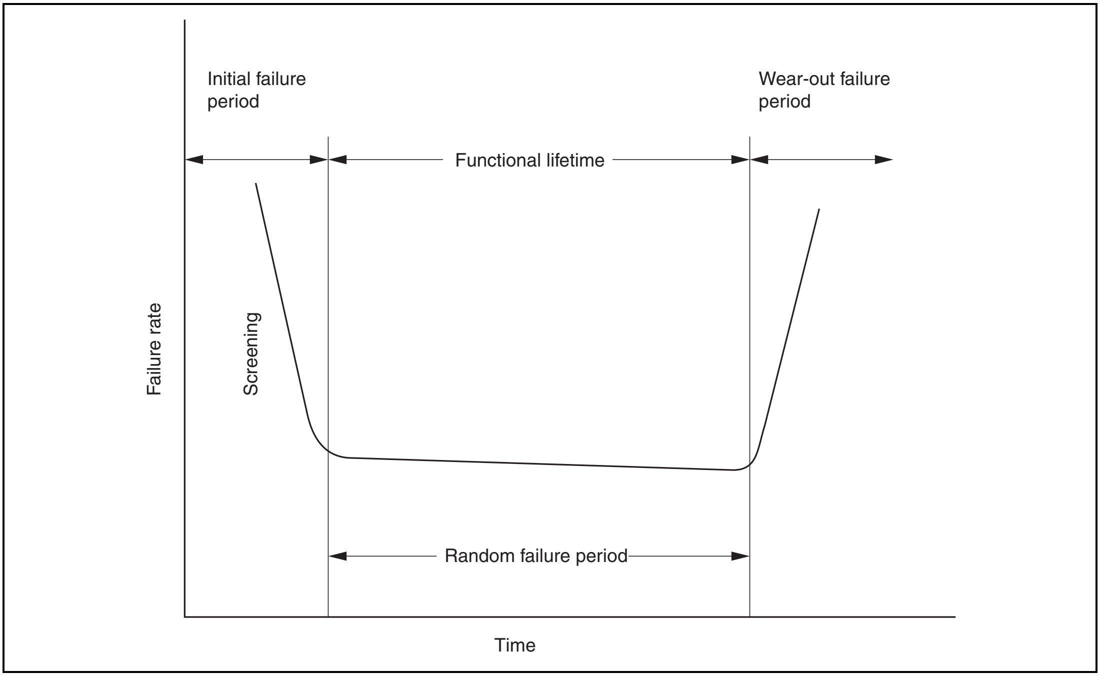


Figure 32.1 Failure Rate Curve (Bathtub Curve)

The reliability of products is estimated on the assumption that products developed for the automotive sector are used in a tougher environment than products for the consumer and industrial sectors. The representative failure phenomena of semiconductor devices, such as the dielectric breakdown of oxide films and electromigration in wiring, constitute wear-out failures. The stress factors in such failures are the voltage, current, and temperature applied to devices while they are in use. Since the temperature range for the guaranteed operation of products for use in automobiles is conventionally –40°C to 125°C, their reliability in terms of the above failure phenomena has to be confirmed by accelerated life testing at all temperatures in this range. Operation at temperatures in excess of 125°C leads to failure within a short time, since high temperatures induce failures in semiconductor devices.

Section 33 List of Registers

The register list gives information on the on-chip I/O registers and is configured as described below.

Register Addresses (grouped by module name, in ordered of the corresponding section numbers):

- Descriptions by functional module, in order of the corresponding section numbers
- Access to reserved addresses which are not described in this list is disabled.
- When registers consist of 16 or 32 bits, the addresses of the MSBs are given, on the presumption of a big-endian system.

List of Register Bits:

- Bit configuration in each register is described in the same fashion as in Register Addresses (grouped by module name, in ordered of the corresponding section numbers).
- The character “-” in the bit name cell indicates a reserved bit.
- The blanks in the bit name cells in a whole line indicates that the corresponding register is allocated to a specific counter or data.

Register States in Each Operating Mode:

- Register states are described in the same order as the Register Addresses (by functional module, in order of the corresponding section numbers).
- For the initial state of each bit, refer to the description of the register in the corresponding section.
- The register states described are for the basic operating modes. If there is a specific reset for an on-chip module, refer to the section on that on-chip module.

33.1 Register Addresses (grouped by module name, in ordered of the corresponding section numbers)

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
INTC	Interrupt control register 0	ICR0	16	H'FFFE0800	16, 32
	Interrupt control register 1	ICR1	16	H'FFFE0802	16, 32
	IRQ interrupt request register	IRQRR	16	H'FFFE0806	8, 16, 32
	Bank control register	IBCR	16	H'FFFE080C	16, 32
	Bank number register	IBNR	16	H'FFFE080E	16, 32
	Software interrupt register 1	SINTR1	8	H'FFFE0810	8, 16, 32
	Software interrupt register 2	SINTR2	8	H'FFFE0811	8, 16, 32
	Software interrupt register 3	SINTR3	8	H'FFFE0812	8, 16, 32
	Software interrupt register 4	SINTR4	8	H'FFFE0813	8, 16, 32
	Software interrupt register 5	SINTR5	8	H'FFFE0814	8, 16, 32
	Software interrupt register 6	SINTR6	8	H'FFFE0815	8, 16, 32
	Software interrupt register 7	SINTR7	8	H'FFFE0816	8, 16, 32
	Software interrupt register 8	SINTR8	8	H'FFFE0817	8, 16, 32
	Interrupt priority register 01	IPR01	16	H'FFFE0818	16, 32
	Interrupt priority register 02	IPR02	16	H'FFFE081A	16, 32
	Software interrupt register 9	SINTR9	8	H'FFFE0828	8, 16, 32
	Software interrupt register 10	SINTR10	8	H'FFFE0829	8, 16, 32
	Software interrupt register 11	SINTR11	8	H'FFFE082A	8, 16, 32
	Software interrupt register 12	SINTR12	8	H'FFFE082B	8, 16, 32
	Software interrupt register 13	SINTR13	8	H'FFFE082C	8, 16, 32
	Software interrupt register 14	SINTR14	8	H'FFFE082D	8, 16, 32
	Software interrupt register 15	SINTR15	8	H'FFFE082E	8, 16, 32
	Interrupt priority register 03	IPR03	16	H'FFFE0C00	16, 32
	Interrupt priority register 04	IPR04	16	H'FFFE0C02	16, 32
	Interrupt priority register 05	IPR05	16	H'FFFE0C04	16, 32
	Interrupt priority register 06	IPR06	16	H'FFFE0C06	16, 32
	Interrupt priority register 07	IPR07	16	H'FFFE0C08	16, 32

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
INTC	Interrupt priority register 08	IPR08	16	H'FFFE0C0A	16, 32
	Interrupt priority register 09	IPR09	16	H'FFFE0C0C	16, 32
	Interrupt priority register 10	IPR10	16	H'FFFE0C0E	16, 32
	Interrupt priority register 11	IPR11	16	H'FFFE0C10	16, 32
	Interrupt priority register 12	IPR12	16	H'FFFE0C12	16, 32
	Interrupt priority register 13	IPR13	16	H'FFFE0C14	16, 32
	Interrupt priority register 14	IPR14	16	H'FFFE0C16	16, 32
	Interrupt priority register 15	IPR15	16	H'FFFE0C18	16, 32
	Interrupt priority register 16	IPR16	16	H'FFFE0C1A	16, 32
	Interrupt priority register 17	IPR17	16	H'FFFE0C1C	16, 32
	Interrupt priority register 18	IPR18	16	H'FFFE0C1E	16, 32
	Interrupt priority register 19	IPR19	16	H'FFFE0C20	16, 32
	Interrupt priority register 20	IPR20	16	H'FFFE0C22	16, 32
	Interrupt priority register 21	IPR21	16	H'FFFE0C24	16, 32
	Interrupt priority register 22	IPR22	16	H'FFFE0C26	16, 32
	Interrupt priority register 23	IPR23	16	H'FFFE0C28	16, 32
	Interrupt priority register 24	IPR24	16	H'FFFE0C2A	16, 32
	Interrupt priority register 25	IPR25	16	H'FFFE0C2C	16, 32
	Interrupt priority register 26	IPR26	16	H'FFFE0C2E	16, 32
	Interrupt priority register 27	IPR27	16	H'FFFE0C30	16, 32
	Interrupt priority register 28	IPR28	16	H'FFFE0C32	16, 32
	Interrupt priority register 29	IPR29	16	H'FFFE0C34	16, 32
	Interrupt priority register 30	IPR30	16	H'FFFE0C70	16, 32
UBC	Break address register 0	BAR0	32	H'FFFC0400	32
	Break address mask register 0	BAMR0	32	H'FFFC0404	32
	Break bus cycle register 0	BBR0	16	H'FFFC04A0	16
	Break address register 1	BAR1	32	H'FFFC0410	32
	Break address mask register 1	BAMR1	32	H'FFFC0414	32
	Break bus cycle register 1	BBR1	16	H'FFFC04B0	16

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
UBC	Break address register 2	BAR2	32	H'FFFC0420	32
	Break address mask register 2	BAMR2	32	H'FFFC0424	32
	Break bus cycle register 2	BBR2	16	H'FFFC04A4	16
	Break address register 3	BAR3	32	H'FFFC0430	32
	Break address mask register 3	BAMR3	32	H'FFFC0434	32
	Break bus cycle register 3	BBR3	16	H'FFFC04B4	16
	Break control register	BRCR	32	H'FFFC04C0	8, 32
	Break address register 4	BAR4	32	H'FFFC0500	32
	Break address mask register 4	BAMR4	32	H'FFFC0504	32
	Break bus cycle register 4	BBR4	16	H'FFFC05A0	16
	Break address register 5	BAR5	32	H'FFFC0510	32
	Break address mask register 5	BAMR5	32	H'FFFC0514	32
	Break bus cycle register 5	BBR5	16	H'FFFC05B0	16
	Break address register 6	BAR6	32	H'FFFC0520	32
	Break address mask register 6	BAMR6	32	H'FFFC0524	32
	Break bus cycle register 6	BBR6	16	H'FFFC05A4	16
	Break address register 7	BAR7	32	H'FFFC0530	32
	Break address mask register 7	BAMR7	32	H'FFFC0534	32
	Break bus cycle register 7	BBR7	16	H'FFFC05B4	16
	Break control register 1	BRCR1	32	H'FFFC05C0	8, 32
BSC	CS0 space bus control register	CS0BCR	32	H'FFFC0004	32
	CS1 space bus control register	CS1BCR	32	H'FFFC0008	32
	CS2 space bus control register	CS2BCR	32	H'FFFC000C	32
	CS3 space bus control register	CS3BCR	32	H'FFFC0010	32
	CS0 space wait control register	CS0WCR	32	H'FFFC0028	32
	CS1 space wait control register	CS1WCR	32	H'FFFC002C	32
	CS2 space wait control register	CS2WCR	32	H'FFFC0030	32
	CS3 space wait control register	CS3WCR	32	H'FFFC0034	32
DMAC	DMA source address register 0	SAR0	32	H'FFFE1000	16, 32
	DMA destination address register 0	DAR0	32	H'FFFE1004	16, 32

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
DMAC	DMA transfer count register 0	DMATCR0	32	H'FFFE1008	16, 32
	DMA channel control register 0	CHCR0	32	H'FFFE100C	8, 16, 32
	DMA channel flag bit register 0	CHFR0	8	H'FFFE108C	8
	DMA TE flag mask setting register 0	TEMSK0	16	H'FFFE108E	8, 16
	DMA reload SAR0	RSAR0	32	H'FFFE1100	16, 32
	DMA reload DAR0	RDAR0	32	H'FFFE1104	16, 32
	DMA reload DMATCR0	RDMATCR0	32	H'FFFE1108	16, 32
	DMA source address register 1	SAR1	32	H'FFFE1010	16, 32
	DMA destination address register 1	DAR1	32	H'FFFE1014	16, 32
	DMA transfer count register 1	DMATCR1	32	H'FFFE1018	16, 32
	DMA channel control register 1	CHCR1	32	H'FFFE101C	8, 16, 32
	DMA channel flag bit register 1	CHFR1	8	H'FFFE109C	8
	DMA TE flag mask setting register 1	TEMSK1	16	H'FFFE109E	8, 16
	DMA reload SAR1	RSAR1	32	H'FFFE1110	16, 32
	DMA reload DAR1	RDAR1	32	H'FFFE1114	16, 32
	DMA reload DMATCR1	RDMATCR1	32	H'FFFE1118	16, 32
	DMA source address register 2	SAR2	32	H'FFFE1020	16, 32
	DMA destination address register 2	DAR2	32	H'FFFE1024	16, 32
	DMA transfer count register 2	DMATCR2	32	H'FFFE1028	16, 32
	DMA channel control register 2	CHCR2	32	H'FFFE102C	8, 16, 32
	DMA channel flag bit register 2	CHFR2	8	H'FFFE10AC	8
	DMA TE flag mask setting register 2	TEMSK2	16	H'FFFE10AE	8, 16
	DMA reload SAR2	RSAR2	32	H'FFFE1120	16, 32
	DMA reload DAR2	RDAR2	32	H'FFFE1124	16, 32
	DMA reload DMATCR2	RDMATCR2	32	H'FFFE1128	16, 32
	DMA source address register 3	SAR3	32	H'FFFE1030	16, 32
	DMA destination address register 3	DAR3	32	H'FFFE1034	16, 32

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
DMAC	DMA transfer count register 3	DMATCR3	32	H'FFFE1038	16, 32
	DMA channel control register 3	CHCR3	32	H'FFFE103C	8, 16, 32
	DMA channel flag bit register 3	CHFR3	8	H'FFFE10BC	8
	DMA TE flag mask setting register 3	TEMSK3	16	H'FFFE10BE	8, 16
	DMA reload SAR3	RSAR3	32	H'FFFE1130	16, 32
	DMA reload DAR3	RDAR3	32	H'FFFE1134	16, 32
	DMA reload DMATCR3	RDMATCR3	32	H'FFFE1138	16, 32
	DMA source address register 4	SAR4	32	H'FFFE1040	16, 32
	DMA destination address register 4	DAR4	32	H'FFFE1044	16, 32
	DMA transfer count register 4	DMATCR4	32	H'FFFE1048	16, 32
	DMA channel control register 4	CHCR4	32	H'FFFE104C	8, 16, 32
	DMA channel flag bit register 4	CHFR4	8	H'FFFE10CC	8
	DMA TE flag mask setting register 4	TEMSK4	16	H'FFFE10CE	8, 16
	DMA reload SAR4	RSAR4	32	H'FFFE1140	16, 32
	DMA reload DAR4	RDAR4	32	H'FFFE1144	16, 32
	DMA reload DMATCR4	RDMATCR4	32	H'FFFE1148	16, 32
	DMA address reload count register 4	ARCR4	16	H'FFFE114C	16, 32
	DMA reload SARCR4	RARCR4	16	H'FFFE114E	16
	DMA source address register 5	SAR5	32	H'FFFE1050	16, 32
	DMA destination address register 5	DAR5	32	H'FFFE1054	16, 32
	DMA transfer count register 5	DMATCR5	32	H'FFFE1058	16, 32
	DMA channel control register 5	CHCR5	32	H'FFFE105C	8, 16, 32
	DMA channel flag bit register 5	CHFR5	8	H'FFFE10DC	8
	DMA TE flag mask setting register 5	TEMSK5	16	H'FFFE10DE	8, 16
	DMA reload SAR5	RSAR5	32	H'FFFE1150	16, 32
	DMA reload DAR5	RDAR5	32	H'FFFE1154	16, 32
	DMA reload DMATCR5	RDMATCR5	32	H'FFFE1158	16, 32

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
DMAC	DMA address reload count register 5	ARCR5	16	H'FFFE115C	16, 32
	DMA reload SARCR5	RARCR5	16	H'FFFE115E	16
	DMA source address register 6	SAR6	32	H'FFFE1060	16, 32
	DMA destination address register 6	DAR6	32	H'FFFE1064	16, 32
	DMA transfer count register 6	DMATCR6	32	H'FFFE1068	16, 32
	DMA channel control register 6	CHCR6	32	H'FFFE106C	8, 16, 32
	DMA channel flag bit register 6	CHFR6	8	H'FFFE10EC	8
	DMA TE flag mask setting register 6	TEMSK6	16	H'FFFE10EE	8, 16
	DMA reload SAR6	RSAR6	32	H'FFFE1160	16, 32
	DMA reload DAR6	RDAR6	32	H'FFFE1164	16, 32
	DMA reload DMATCR6	RDMATCR6	32	H'FFFE1168	16, 32
	DMA address reload count register 6	ARCR6	16	H'FFFE116C	16, 32
	DMA reload SARCR6	RARCR6	16	H'FFFE116E	16
	DMA source address register 7	SAR7	32	H'FFFE1070	16, 32
	DMA destination address register 7	DAR7	32	H'FFFE1074	16, 32
	DMA transfer count register 7	DMATCR7	32	H'FFFE1078	16, 32
	DMA channel control register 7	CHCR7	32	H'FFFE107C	8, 16, 32
	DMA channel flag bit register 7	CHFR7	8	H'FFFE10FC	8
	DMA TE flag mask setting register 7	TEMSK7	16	H'FFFE10FE	8, 16
	DMA reload SAR7	RSAR7	32	H'FFFE1170	16, 32
	DMA reload DAR7	RDAR7	32	H'FFFE1174	16, 32
	DMA reload DMATCR7	RDMATCR7	32	H'FFFE1178	16, 32
	DMA address reload count register 7	ARCR7	16	H'FFFE117C	16, 32
	DMA reload SARCR7	RARCR7	16	H'FFFE117E	16
	DMA operation register	DMAOR	16	H'FFFE1200	8, 16
	DMA operation flag bit register	DMAFR	8	H'FFFE1204	8

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
DMAC	DMA extension resource selector 0	DMARS0	16	H'FFFE1300	8, 16
	DMA extension resource selector 1	DMARS1	16	H'FFFE1304	8, 16
	DMA extension resource selector 2	DMARS2	16	H'FFFE1308	8, 16
	DMA extension resource selector 3	DMARS3	16	H'FFFE130C	8, 16
A-DMAC	A-DMAC operation register	ADMAOR	8	H'FFFE6000	8
	A-DMAC alias base register	ADMAABR	8	H'FFFE6002	8
	A-DMAC interrupt control register 0	ADMAIE0	8	H'FFFE6010	8
	A-DMAC interrupt control register 1	ADMAIE1	8	H'FFFE6011	8
	A-DMAC interrupt control register 2	ADMAIE2	8	H'FFFE6012	8
	A-DMAC interrupt control register 3	ADMAIE3	8	H'FFFE6013	8
	A-DMAC interrupt control register 4	ADMAIE4	8	H'FFFE6014	8
	A-DMAC interrupt control register 5	ADMAIE5	8	H'FFFE6015	8
	A-DMAC interrupt control register 6	ADMAIE6	8	H'FFFE6016	8
	A-DMAC interrupt control register 7	ADMAIE7	8	H'FFFE6017	8
	A-DMAC interrupt control register 8	ADMAIE8	8	H'FFFE6018	8
	A-DMAC interrupt control register 9	ADMAIE9	8	H'FFFE6019	8
	A-DMAC data valid register 0	ADMADV0	8	H'FFFE6020	8
	A-DMAC data valid register 1	ADMADV1	8	H'FFFE6021	8
	A-DMAC data valid register 2	ADMADV2	8	H'FFFE6022	8
	A-DMAC data valid register 3	ADMADV3	8	H'FFFE6023	8
	A-DMAC data valid register 4	ADMADV4	8	H'FFFE6024	8

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
A-DMAC	A-DMAC data valid register 5	ADMADV5	8	H'FFFE6025	8
	A-DMAC data valid register 6	ADMADV6	8	H'FFFE6026	8
	A-DMAC transfer end register 0	ADMATE0	8	H'FFFE6030	8
	A-DMAC transfer end register 1	ADMATE1	8	H'FFFE6031	8
	A-DMAC transfer end register 2	ADMATE2	8	H'FFFE6032	8
	A-DMAC enable register 0	ADMADE0	8	H'FFFE6040	8
	A-DMAC enable register 1	ADMADE1	8	H'FFFE6041	8
	A-DMAC enable register 2	ADMADE2	8	H'FFFE6042	8
	A-DMAC enable register 3	ADMADE3	8	H'FFFE6043	8
	A-DMAC enable register 4	ADMADE4	8	H'FFFE6044	8
	A-DMAC enable register 5	ADMADE5	8	H'FFFE6045	8
	A-DMAC enable register 6	ADMADE6	8	H'FFFE6046	8
	A-DMAC enable register 7	ADMADE7	8	H'FFFE6047	8
	A-DMAC transfer mode register 0	ADMAMODE0	8	H'FFFE6050	8
	A-DMAC transfer mode register 1	ADMAMODE1	8	H'FFFE6051	8
	A-DMAC transfer mode register 2	ADMAMODE2	8	H'FFFE6052	8
	A-DMAC transfer mode register 3	ADMAMODE3	8	H'FFFE6053	8
	A-DMAC transfer count register 0	ADMATCR0	16	H'FFFE6060	16
	A-DMAC reload transfer count register 0	ADMARTCR0	16	H'FFFE6062	16
	A-DMAC transfer count register 1	ADMATCR1	16	H'FFFE6064	16
	A-DMAC reload transfer count register 1	ADMARTCR1	16	H'FFFE6066	16
	A-DMAC transfer count register 56	ADMATCR56	16	H'FFFE6070	16
	A-DMAC transfer count register 57	ADMATCR57	16	H'FFFE6072	16
	A-DMAC transfer count register 58	ADMATCR58	16	H'FFFE6074	16
	A-DMAC transfer count register 59	ADMATCR59	16	H'FFFE6076	16
	A-DMAC transfer count register 60	ADMATCR60	16	H'FFFE6078	16

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
A-DMAC	A-DMAC transfer count register 61	ADMATCR61	16	H'FFFE607A	16
	A-DMAC transfer count register 62	ADMATCR62	16	H'FFFE607C	16
	A-DMAC transfer count register 63	ADMATCR63	16	H'FFFE607E	16
	A-DMAC transfer count register 64	ADMATCR64	16	H'FFFE6080	16
	A-DMAC transfer count register 65	ADMATCR65	16	H'FFFE6082	16
	A-DMAC transfer count register 66	ADMATCR66	16	H'FFFE6084	16
	A-DMAC transfer count register 67	ADMATCR67	16	H'FFFE6086	16
	A-DMAC transfer count register 68	ADMATCR68	16	H'FFFE6088	16
	A-DMAC transfer count register 69	ADMATCR69	16	H'FFFE608A	16
	A-DMAC transfer count register 70	ADMATCR70	16	H'FFFE608C	16
	A-DMAC transfer count register 71	ADMATCR71	16	H'FFFE608E	16
	A-DMAC alias pointer register 0	ADMAAR0	16	H'FFFE6090	16
	A-DMAC reload alias pointer register 0	ADMARAR0	16	H'FFFE6092	16
	A-DMAC alias pointer register 1	ADMAAR1	16	H'FFFE6094	16
	A-DMAC reload alias pointer register 1	ADMARAR1	16	H'FFFE6096	16
	A-DMAC alias pointer register 56	ADMAAR56	16	H'FFFE60A0	16
	A-DMAC alias pointer register 57	ADMAAR57	16	H'FFFE60A2	16
	A-DMAC alias pointer register 58	ADMAAR58	16	H'FFFE60A4	16
	A-DMAC alias pointer register 59	ADMAAR59	16	H'FFFE60A6	16
	A-DMAC alias pointer register 60	ADMAAR60	16	H'FFFE60A8	16
	A-DMAC alias pointer register 61	ADMAAR61	16	H'FFFE60AA	16
	A-DMAC alias pointer register 62	ADMAAR62	16	H'FFFE60AC	16

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
A-DMAC	A-DMAC alias pointer register 63	ADMAAR63	16	H'FFFE60AE	16
	A-DMAC alias pointer register 64	ADMAAR64	16	H'FFFE60B0	16
	A-DMAC alias pointer register 65	ADMAAR65	16	H'FFFE60B2	16
	A-DMAC alias pointer register 66	ADMAAR66	16	H'FFFE60B4	16
	A-DMAC alias pointer register 67	ADMAAR67	16	H'FFFE60B6	16
	A-DMAC alias pointer register 68	ADMAAR68	16	H'FFFE60B8	16
	A-DMAC alias pointer register 69	ADMAAR69	16	H'FFFE60BA	16
	A-DMAC alias pointer register 70	ADMAAR70	16	H'FFFE60BC	16
	A-DMAC alias pointer register 71	ADMAAR71	16	H'FFFE60BE	16
	A-DMAC buffer register 2	ADMABUF2	32	H'FFFE60C0	32
	A-DMAC buffer register 3	ADMABUF3	32	H'FFFE60C4	32
	A-DMAC buffer register 4	ADMABUF4	32	H'FFFE60C8	32
	A-DMAC buffer register 5	ADMABUF5	32	H'FFFE60CC	32
	A-DMAC buffer register 6	ADMABUF6	32	H'FFFE60D0	32
	A-DMAC buffer register 7	ADMABUF7	32	H'FFFE60D4	32
	A-DMAC receive wait register 0	ADMARVPR0	16	H'FFFE60E0	8, 16
	A-DMAC receive wait register 1	ADMARVPR1	16	H'FFFE60E2	8, 16
	A-DMAC receive wait register 2	ADMARVPR2	16	H'FFFE60E4	8, 16
	A-DMAC receive wait register 3	ADMARVPR3	16	H'FFFE60E6	8, 16
	A-DMAC receive wait register 4	ADMARVPR4	16	H'FFFE60E8	8, 16
	A-DMAC receive wait register 5	ADMARVPR5	16	H'FFFE60EA	8, 16
	A-DMAC receive wait register 6	ADMARVPR6	16	H'FFFE60EC	8, 16
	A-DMAC receive wait register 7	ADMARVPR7	16	H'FFFE60EE	8, 16
	A-DMAC transmit wait register 0	ADMATVPR0	16	H'FFFE60F0	8, 16
	A-DMAC transmit wait register 1	ADMATVPR1	16	H'FFFE60F2	8, 16
	A-DMAC transmit wait register 2	ADMATVPR2	16	H'FFFE60F4	8, 16
	A-DMAC transmit wait register 3	ADMATVPR3	16	H'FFFE60F6	8, 16
	A-DMAC transmit wait register 4	ADMATVPR4	16	H'FFFE60F8	8, 16
	A-DMAC transmit wait register 5	ADMATVPR5	16	H'FFFE60FA	8, 16
	A-DMAC transmit wait register 6	ADMATVPR6	16	H'FFFE60FC	8, 16
	A-DMAC transmit wait register 7	ADMATVPR7	16	H'FFFE60FE	8, 16

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
A-DMAC	A-DMAC FlexRay receive wait register 0	ADMAFRWR0	32	H'FFFC2800	8, 16, 32
	A-DMAC FlexRay receive wait register 1	ADMAFRWR1	32	H'FFFC2804	8, 16, 32
	A-DMAC FlexRay receive wait register 2	ADMAFRWR2	32	H'FFFC2808	8, 16, 32
	A-DMAC FlexRay receive wait register 3	ADMAFRWR3	32	H'FFFC280C	8, 16, 32
	A-DMAC FlexRay last message buffer register	ADMAFRLMB	8	H'FFFC2810	8, 16, 32
	A-DMAC FlexRay general control register	ADMAFRGENCTR	8	H'FFFC2811	8
	A-DMAC FlexRay control register	ADMAFRCTR	8	H'FFFC2812	8, 16
	A-DMAC FlexRay FIFO control register	ADMAFRFCTR	8	H'FFFC2813	8
	A-DMAC FlexRay transmit status register	ADMAFRTRSTAT	8	H'FFFC2814	8, 16, 32
	A-DMAC FlexRay status register	ADMAFRSTAT	8	H'FFFC2816	8, 16
	A-DMAC FlexRay FIFO status register	ADMAFRFSTAT	8	H'FFFC2817	8
ATU-III	ATU-III master enable register	ATUENR	16	H'FFFFF000	8, 16
	Clock bus control register	CBCNT	8	H'FFFFF002	8
	Noise cancellation mode register	NCMR	8	H'FFFFF003	8
	Prescaler register 0	PSCR0	16	H'FFFFF100	16
	Prescaler register 1	PSCR1	16	H'FFFFF102	16
	Prescaler register 2	PSCR2	16	H'FFFFF104	16
	Prescaler register 3	PSCR3	16	H'FFFFF106	16
	Timer control register A	TCRA	8	H'FFFFF202	8
	Timer I/O control register 1A	TIOR1A	16	H'FFFFF204	8, 16
	Timer I/O control register 2A	TIOR2A	16	H'FFFFF206	8, 16
	Timer status register A	TSRA	8	H'FFFFF208	8
	Timer interrupt enable register A	TIERA	8	H'FFFFF209	8

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
ATU-III	Noise canceler mode register 1A	NCMCR1A	8	H'FFFFF20A	8
	Noise canceler counter A0	NCNTA0	8	H'FFFFF210	8
	Noise canceler register A0	NCRA0	8	H'FFFFF211	8
	Noise canceler counter A1	NCNTA1	8	H'FFFFF212	8
	Noise canceler register A1	NCRA1	8	H'FFFFF213	8
	Noise canceler counter A2	NCNTA2	8	H'FFFFF214	8
	Noise canceler register A2	NCRA2	8	H'FFFFF215	8
	Noise canceler counter A3	NCNTA3	8	H'FFFFF216	8
	Noise canceler register A3	NCRA3	8	H'FFFFF217	8
	Noise canceler counter A4	NCNTA4	8	H'FFFFF218	8
	Noise canceler register A4	NCRA4	8	H'FFFFF219	8
	Noise canceler counter A5	NCNTA5	8	H'FFFFF21A	8
	Noise canceler register A5	NCRA5	8	H'FFFFF21B	8
	Free-running counter A	TCNTA	32	H'FFFFF220	32
	Input capture register A0	ICRA0	32	H'FFFFF228	32
	Input capture register A1	ICRA1	32	H'FFFFF22C	32
	Input capture register A2	ICRA2	32	H'FFFFF230	32
	Input capture register A3	ICRA3	32	H'FFFFF234	32
	Input capture register A4	ICRA4	32	H'FFFFF238	32
	Input capture register A5	ICRA5	32	H'FFFFF23C	32
	Timer control register B	TCRB	8	H'FFFFF304	8
	Timer I/O control register B	TIORB	8	H'FFFFF305	8
	Timer status register B	TSRB	8	H'FFFFF306	8
	Timer interrupt enable register B	TIERB	8	H'FFFFF307	8
	Edge interval measuring counter B0	TCNTB0	32	H'FFFFF310	32
	Input capture register B0	ICRB0	32	H'FFFFF314	32
	Output compare register B0	OCRB0	32	H'FFFFF318	32
	Event counter B1	TCNTB1	8	H'FFFFF31C	8
	Output compare register B1	OCRB1	8	H'FFFFF31D	8

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
ATU-III	Input capture register B1	ICRB1	32	H'FFFFF320	32
	Input capture register B2	ICRB2	32	H'FFFFF324	32
	Load register B	LDB	32	H'FFFFF330	32
	Reload register B	RLDB	32	H'FFFFF334	32
	Pulse interval multiplier register	PIMR	16	H'FFFFF338	16
	Reloadable counter B2	TCNTB2	32	H'FFFFF33C	32
	Multiplied clock counter B6	TCNTB6	32	H'FFFFF340	32
	Output compare register B6	OCRB6	32	H'FFFFF344	32
	Output compare register B7	OCRB7	32	H'FFFFF348	32
	Correcting event counter B3	TCNTB3	32	H'FFFFF350	32
	Multiplied-and-corrected clock counter B4	TCNTB4	32	H'FFFFF354	32
	Generating frequency-multiplied corrected clock counter B5	TCNTB5	32	H'FFFFF358	32
	Correcting counter clearing register B	TCCLRB	32	H'FFFFF35C	32
	Timer start register C	TSTRC	8	H'FFFFF400	8
	Noise canceler control register C0	NCCRC0	8	H'FFFFF402	8
	Noise canceler control register C1	NCCRC1	8	H'FFFFF403	8
	Noise canceler control register C2	NCCRC2	8	H'FFFFF404	8
	Noise canceler control register C3	NCCRC3	8	H'FFFFF405	8
	Noise canceler control register C4	NCCRC4	8	H'FFFFF406	8
	Noise canceler counter C00	NCNTC00	8	H'FFFFF410	8
	Noise canceler counter C01	NCNTC01	8	H'FFFFF411	8
	Noise canceler counter C02	NCNTC02	8	H'FFFFF412	8
	Noise canceler counter C03	NCNTC03	8	H'FFFFF413	8
	Noise cancel register C00	NCRC00	8	H'FFFFF414	8

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
ATU-III	Noise cancel register C01	NCRC01	8	H'FFFFF415	8
	Noise cancel register C02	NCRC02	8	H'FFFFF416	8
	Noise cancel register C03	NCRC03	8	H'FFFFF417	8
	Noise canceler counter C10	NCNTC10	8	H'FFFFF418	8
	Noise canceler counter C11	NCNTC11	8	H'FFFFF419	8
	Noise canceler counter C12	NCNTC12	8	H'FFFFF41A	8
	Noise canceler counter C13	NCNTC13	8	H'FFFFF41B	8
	Noise cancel register C10	NCRC10	8	H'FFFFF41C	8
	Noise cancel register C11	NCRC11	8	H'FFFFF41D	8
	Noise cancel register C12	NCRC12	8	H'FFFFF41E	8
	Noise cancel register C13	NCRC13	8	H'FFFFF41F	8
	Noise canceler counter C20	NCNTC20	8	H'FFFFF420	8
	Noise canceler counter C21	NCNTC21	8	H'FFFFF421	8
	Noise canceler counter C22	NCNTC22	8	H'FFFFF422	8
	Noise canceler counter C23	NCNTC23	8	H'FFFFF423	8
	Noise cancel register C20	NCRC20	8	H'FFFFF424	8
	Noise cancel register C21	NCRC21	8	H'FFFFF425	8
	Noise cancel register C22	NCRC22	8	H'FFFFF426	8
	Noise cancel register C23	NCRC23	8	H'FFFFF427	8
	Noise canceler counter C30	NCNTC30	8	H'FFFFF428	8
	Noise canceler counter C31	NCNTC31	8	H'FFFFF429	8
	Noise canceler counter C32	NCNTC32	8	H'FFFFF42A	8
	Noise canceler counter C33	NCNTC33	8	H'FFFFF42B	8
	Noise cancel register C30	NCRC30	8	H'FFFFF42C	8
	Noise cancel register C31	NCRC31	8	H'FFFFF42D	8
	Noise cancel register C32	NCRC32	8	H'FFFFF42E	8
	Noise cancel register C33	NCRC33	8	H'FFFFF42F	8
	Noise canceler counter C40	NCNTC40	8	H'FFFFF430	8
	Noise canceler counter C41	NCNTC41	8	H'FFFFF431	8
	Noise canceler counter C42	NCNTC42	8	H'FFFFF432	8
	Noise canceler counter C43	NCNTC43	8	H'FFFFF433	8

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
ATU-III	Noise cancel register C40	NCRC40	8	H'FFFFF434	8
	Noise cancel register C41	NCRC41	8	H'FFFFF435	8
	Noise cancel register C42	NCRC42	8	H'FFFFF436	8
	Noise cancel register C43	NCRC43	8	H'FFFFF437	8
	Timer control register C0	TCRC0	8	H'FFFFF440	8
	Timer interrupt enable register C0	TIERC0	8	H'FFFFF441	8
	Timer I/O control register C0	TIORC0	16	H'FFFFF442	8, 16
	Timer status register C0	TSRC0	8	H'FFFFF444	8
	General register C00	GRC00	32	H'FFFFF448	32
	General register C01	GRC01	32	H'FFFFF44C	32
	General register C02	GRC02	32	H'FFFFF450	32
	General register C03	GRC03	32	H'FFFFF454	32
	Timer counter C0	TCNTC0	32	H'FFFFF458	32
	Timer control register C1	TCRC1	8	H'FFFFF460	8
	Timer interrupt enable register C1	TIERC1	8	H'FFFFF461	8
	Timer I/O control register C1	TIORC1	16	H'FFFFF462	8, 16
	Timer status register C1	TSRC1	8	H'FFFFF464	8
	General register C10	GRC10	32	H'FFFFF468	32
	General register C11	GRC11	32	H'FFFFF46C	32
	General register C12	GRC12	32	H'FFFFF470	32
	General register C13	GRC13	32	H'FFFFF474	32
	Timer counter C1	TCNTC1	32	H'FFFFF478	32
	Timer control register C2	TCRC2	8	H'FFFFF480	8
	Timer interrupt enable register C2	TIERC2	8	H'FFFFF481	8
	Timer I/O control register C2	TIORC2	16	H'FFFFF482	8, 16
	Timer status register C2	TSRC2	8	H'FFFFF484	8
	General register C20	GRC20	32	H'FFFFF488	32
	General register C21	GRC21	32	H'FFFFF48C	32
	General register C22	GRC22	32	H'FFFFF490	32

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
ATU-III	General register C23	GRC23	32	H'FFFFF494	32
	Timer counter C2	TCNTC2	32	H'FFFFF498	32
	Timer control register C3	TCRC3	8	H'FFFFF4A0	8
	Timer interrupt enable register C3	TIERC3	8	H'FFFFF4A1	8
	Timer I/O control register C3	TIORC3	16	H'FFFFF4A2	8, 16
	Timer status register C3	TSRC3	8	H'FFFFF4A4	8
	General register C30	GRC30	32	H'FFFFF4A8	32
	General register C31	GRC31	32	H'FFFFF4AC	32
	General register C32	GRC32	32	H'FFFFF4B0	32
	General register C33	GRC33	32	H'FFFFF4B4	32
	Timer counter C3	TCNTC3	32	H'FFFFF4B8	32
	Timer control register C4	TCRC4	8	H'FFFFF4C0	8
	Timer interrupt enable register C4	TIERC4	8	H'FFFFF4C1	8
	Timer I/O control register C4	TIORC4	16	H'FFFFF4C2	8, 16
	Timer status register C4	TSRC4	8	H'FFFFF4C4	8
	General register C40	GRC40	32	H'FFFFF4C8	32
	General register C41	GRC41	32	H'FFFFF4CC	32
	General register C42	GRC42	32	H'FFFFF4D0	32
	General register C43	GRC43	32	H'FFFFF4D4	32
	Timer counter C4	TCNTC4	32	H'FFFFF4D8	32
	Timer start register D	TSTRD	8	H'FFFFF500	8
	Timer counter 1D0	TCNT1D0	32	H'FFFFF520	32
	Timer counter 2D0	TCNT2D0	32	H'FFFFF524	32
	Timer offset base register D0	OSBRD0	32	H'FFFFF528	32
	Timer control register D0	TCRD0	16	H'FFFFF52C	8, 16
	Timer output control register D0	TOCRD0	8	H'FFFFF52E	8
	Compare match pulse output control register D0	CMPOD0	8	H'FFFFF52F	8
	Timer counter 1D1	TCNT1D1	32	H'FFFFF530	32

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
ATU-III	Timer counter 2D1	TCNT2D1	32	H'FFFFF534	32
	Timer offset base register D1	OSBRD1	32	H'FFFFF538	32
	Timer control register D1	TCRD1	16	H'FFFFF53C	8, 16
	Timer output control register D1	TOCRD1	8	H'FFFFF53E	8
	Compare match pulse output control register D1	CMPOD1	8	H'FFFFF53F	8
	Timer counter 1D2	TCNT1D2	32	H'FFFFF540	32
	Timer counter 2D2	TCNT2D2	32	H'FFFFF544	32
	Timer offset base register D2	OSBRD2	32	H'FFFFF548	32
	Timer control register D2	TCRD2	16	H'FFFFF54C	8, 16
	Timer output control register D2	TOCRD2	8	H'FFFFF54E	8
	Timer counter 1D3	TCNT1D3	32	H'FFFFF550	32
	Timer counter 2D3	TCNT2D3	32	H'FFFFF554	32
	Timer offset base register D3	OSBRD3	32	H'FFFFF558	32
	Timer control register D3	TCRD3	16	H'FFFFF55C	8, 16
	Timer output control register D3	TOCRD3	8	H'FFFFF55E	8
	Timer I/O control register 1D0	TIOR1D0	16	H'FFFFF580	16
	Timer I/O control register 2D0	TIOR2D0	16	H'FFFFF582	16
	Down counter starting register D0	DSTRD0	8	H'FFFFF585	8
	Down counter status register D0	DSRD0	8	H'FFFFF587	8
	Down counter control register D0	DCRD0	16	H'FFFFF588	16
	Timer status register D0	TSRD0	16	H'FFFFF58C	16
	Timer interrupt enable register D0	TIERD0	16	H'FFFFF58E	16
	Output compare register D00	OCRD00	32	H'FFFFF590	32
	Output compare register D01	OCRD01	32	H'FFFFF594	32
	Output compare register D02	OCRD02	32	H'FFFFF598	32
	Output compare register D03	OCRD03	32	H'FFFFF59C	32
	General register D00	GRD00	32	H'FFFFF5A0	32
	General register D01	GRD01	32	H'FFFFF5A4	32

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
ATU-III	General register D02	GRD02	32	H'FFFFF5A8	32
	General register D03	GRD03	32	H'FFFFF5AC	32
	Timer down counter D00	DCNTD00	32	H'FFFFF5B0	32
	Timer down counter D01	DCNTD01	32	H'FFFFF5B4	32
	Timer down counter D02	DCNTD02	32	H'FFFFF5B8	32
	Timer down counter D03	DCNTD03	32	H'FFFFF5BC	32
	Timer I/O control register 1D1	TIOR1D1	16	H'FFFFF5C0	16
	Timer I/O control register 2D1	TIOR2D1	16	H'FFFFF5C2	16
	Down counter starting register D1	DSTRD1	8	H'FFFFF5C5	8
	Down counter status register D1	DSRD1	8	H'FFFFF5C7	8
	Down counter control register D1	DCRD1	16	H'FFFFF5C8	16
	Timer status register D1	TSRD1	16	H'FFFFF5CC	16
	Timer interrupt enable register D1	TIERD1	16	H'FFFFF5CE	16
	Output compare register D10	OCRD10	32	H'FFFFF5D0	32
	Output compare register D11	OCRD11	32	H'FFFFF5D4	32
	Output compare register D12	OCRD12	32	H'FFFFF5D8	32
	Output compare register D13	OCRD13	32	H'FFFFF5DC	32
	General register D10	GRD10	32	H'FFFFF5E0	32
	General register D11	GRD11	32	H'FFFFF5E4	32
	General register D12	GRD12	32	H'FFFFF5E8	32
	General register D13	GRD13	32	H'FFFFF5EC	32
	Timer down counter D10	DCNTD10	32	H'FFFFF5F0	32
	Timer down counter D11	DCNTD11	32	H'FFFFF5F4	32
	Timer down counter D12	DCNTD12	32	H'FFFFF5F8	32
	Timer down counter D13	DCNTD13	32	H'FFFFF5FC	32
	Timer I/O control register 1D2	TIOR1D2	16	H'FFFFF600	16
	Timer I/O control register 2D2	TIOR2D2	16	H'FFFFF602	16

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
ATU-III	Down counter starting register D2	DSTRD2	8	H'FFFFF605	8
	Down counter status register D2	DSRD2	8	H'FFFFF607	8
	Down counter control register D2	DCRD2	16	H'FFFFF608	16
	Timer status register D2	TSRD2	16	H'FFFFF60C	16
	Timer interrupt enable register D2	TIERD2	16	H'FFFFF60E	16
	Output compare register D20	OCRD20	32	H'FFFFF610	32
	Output compare register D21	OCRD21	32	H'FFFFF614	32
	Output compare register D22	OCRD22	32	H'FFFFF618	32
	Output compare register D23	OCRD23	32	H'FFFFF61C	32
	General register D20	GRD20	32	H'FFFFF620	32
	General register D21	GRD21	32	H'FFFFF624	32
	General register D22	GRD22	32	H'FFFFF628	32
	General register D23	GRD23	32	H'FFFFF62C	32
	Timer down counter D20	DCNTD20	32	H'FFFFF630	32
	Timer down counter D21	DCNTD21	32	H'FFFFF634	32
	Timer down counter D22	DCNTD22	32	H'FFFFF638	32
	Timer down counter D23	DCNTD23	32	H'FFFFF63C	32
	Timer I/O control register 1D3	TIOR1D3	16	H'FFFFF640	16
	Timer I/O control register 2D3	TIOR2D3	16	H'FFFFF642	16
	Down counter starting register D3	DSTRD3	8	H'FFFFF645	8
	Down counter status register D3	DSRD3	8	H'FFFFF647	8
	Down counter control register D3	DCRD3	16	H'FFFFF648	16
	Timer status register D3	TSRD3	16	H'FFFFF64C	16
	Timer interrupt enable register D3	TIERD3	16	H'FFFFF64E	16

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
ATU-III	Output compare register D30	OCRD30	32	H'FFFFF650	32
	Output compare register D31	OCRD31	32	H'FFFFF654	32
	Output compare register D32	OCRD32	32	H'FFFFF658	32
	Output compare register D33	OCRD33	32	H'FFFFF65C	32
	General register D30	GRD30	32	H'FFFFF660	32
	General register D31	GRD31	32	H'FFFFF664	32
	General register D32	GRD32	32	H'FFFFF668	32
	General register D33	GRD33	32	H'FFFFF66C	32
	Timer down counter D30	DCNTD30	32	H'FFFFF670	32
	Timer down counter D31	DCNTD31	32	H'FFFFF674	32
	Timer down counter D32	DCNTD32	32	H'FFFFF678	32
	Timer down counter D33	DCNTD33	32	H'FFFFF67C	32
	Timer start register E	TSTRE	8	H'FFFFF700	8
	Timer control register E0	TCRE0	8	H'FFFFF800	8
	Timer output control register E0	TOCRE0	8	H'FFFFF801	8
	Timer interrupt enable register E0	TIERE0	8	H'FFFFF802	8
	Reload control register E0	RLDCRE0	8	H'FFFFF803	8
	Timer status register E0	TSRE0	8	H'FFFFF804	8
	Prescaler register E0	PSCRE0	8	H'FFFFF808	8
	Subblock starting register E0	SSTRE0	8	H'FFFFF80C	8
	Cycle-setting register E00	CYLRE00	16	H'FFFFF810	16
	Cycle-setting register E01	CYLRE01	16	H'FFFFF812	16
	Cycle-setting register E02	CYLRE02	16	H'FFFFF814	16
	Cycle-setting register E03	CYLRE03	16	H'FFFFF816	16
	Duty cycle setting register E00	DTRE00	16	H'FFFFF818	16
	Duty cycle setting register E01	DTRE01	16	H'FFFFF81A	16
	Duty cycle setting register E02	DTRE02	16	H'FFFFF81C	16
	Duty cycle setting register E03	DTRE03	16	H'FFFFF81E	16
	Cycle reload register E00	CRLDE00	16	H'FFFFF820	16

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
ATU-III	Cycle reload register E01	CRLDE01	16	H'FFFFF822	16
	Cycle reload register E02	CRLDE02	16	H'FFFFF824	16
	Cycle reload register E03	CRLDE03	16	H'FFFFF826	16
	Duty cycle reload register E00	DRLDE00	16	H'FFFFF828	16
	Duty cycle reload register E01	DRLDE01	16	H'FFFFF82A	16
	Duty cycle reload register E02	DRLDE02	16	H'FFFFF82C	16
	Duty cycle reload register E03	DRLDE03	16	H'FFFFF82E	16
	Timer counter E00	TCNTE00	16	H'FFFFF830	16
	Timer counter E01	TCNTE01	16	H'FFFFF832	16
	Timer counter E02	TCNTE02	16	H'FFFFF834	16
	Timer counter E03	TCNTE03	16	H'FFFFF836	16
	Prescaler channel register E00	PSCCRE00	8	H'FFFFF838	8
	Prescaler channel register E01	PSCCRE01	8	H'FFFFF839	8
	Prescaler channel register E02	PSCCRE02	8	H'FFFFF83A	8
	Prescaler channel register E03	PSCCRE03	8	H'FFFFF83B	8
	Timer control register E1	TCRE1	8	H'FFFFF840	8
	Timer output control register E1	TOCRE1	8	H'FFFFF841	8
	Timer interrupt enable register E1	TIERE1	8	H'FFFFF842	8
	Reload control register E1	RLDCRE1	8	H'FFFFF843	8
	Timer status register E1	TSRE1	8	H'FFFFF844	8
	Prescaler register E1	PSCRE1	8	H'FFFFF848	8
	Subblock starting register E1	SSTRE1	8	H'FFFFF84C	8
	Cycle-setting register E10	CYLRE10	16	H'FFFFF850	16
	Cycle-setting register E11	CYLRE11	16	H'FFFFF852	16
	Cycle-setting register E12	CYLRE12	16	H'FFFFF854	16
	Cycle-setting register E13	CYLRE13	16	H'FFFFF856	16
	Duty cycle setting register E10	DTRE10	16	H'FFFFF858	16
	Duty cycle setting register E11	DTRE11	16	H'FFFFF85A	16
	Duty cycle setting register E12	DTRE12	16	H'FFFFF85C	16

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
ATU-III	Duty cycle setting register E13	DTRE13	16	H'FFFFF85E	16
	Cycle reload register E10	CRLDE10	16	H'FFFFF860	16
	Cycle reload register E11	CRLDE11	16	H'FFFFF862	16
	Cycle reload register E12	CRLDE12	16	H'FFFFF864	16
	Cycle reload register E13	CRLDE13	16	H'FFFFF866	16
	Duty cycle reload register E10	DRLDE10	16	H'FFFFF868	16
	Duty cycle reload register E11	DRLDE11	16	H'FFFFF86A	16
	Duty cycle reload register E12	DRLDE12	16	H'FFFFF86C	16
	Duty cycle reload register E13	DRLDE13	16	H'FFFFF86E	16
	Timer counter E10	TCNTE10	16	H'FFFFF870	16
	Timer counter E11	TCNTE11	16	H'FFFFF872	16
	Timer counter E12	TCNTE12	16	H'FFFFF874	16
	Timer counter E13	TCNTE13	16	H'FFFFF876	16
	Prescaler channel register E10	PSCCRE10	8	H'FFFFF878	8
	Prescaler channel register E11	PSCCRE11	8	H'FFFFF879	8
	Prescaler channel register E12	PSCCRE12	8	H'FFFFF87A	8
	Prescaler channel register E13	PSCCRE13	8	H'FFFFF87B	8
	Timer control register E2	TCRE2	8	H'FFFFF880	8
	Timer output control register E2	TOCRE2	8	H'FFFFF881	8
	Timer interrupt enable register E2	TIERE2	8	H'FFFFF882	8
	Reload control register E2	RLDCRE2	8	H'FFFFF883	8
	Timer status register E2	TSRE2	8	H'FFFFF884	8
	Prescaler register E2	PSCRE2	8	H'FFFFF888	8
	Subblock starting register E2	SSTRE2	8	H'FFFFF88C	8
	Cycle-setting register E20	CYLRE20	16	H'FFFFF890	16
	Cycle-setting register E21	CYLRE21	16	H'FFFFF892	16
	Cycle-setting register E22	CYLRE22	16	H'FFFFF894	16
	Cycle-setting register E23	CYLRE23	16	H'FFFFF896	16
	Duty cycle setting register E20	DTRE20	16	H'FFFFF898	16

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
ATU-III	Duty cycle setting register E21	DTRE21	16	H'FFFFF89A	16
	Duty cycle setting register E22	DTRE22	16	H'FFFFF89C	16
	Duty cycle setting register E23	DTRE23	16	H'FFFFF89E	16
	Cycle reload register E20	CRLDE20	16	H'FFFFF8A0	16
	Cycle reload register E21	CRLDE21	16	H'FFFFF8A2	16
	Cycle reload register E22	CRLDE22	16	H'FFFFF8A4	16
	Cycle reload register E23	CRLDE23	16	H'FFFFF8A6	16
	Duty cycle reload register E20	DRLDE20	16	H'FFFFF8A8	16
	Duty cycle reload register E21	DRLDE21	16	H'FFFFF8AA	16
	Duty cycle reload register E22	DRLDE22	16	H'FFFFF8AC	16
	Duty cycle reload register E23	DRLDE23	16	H'FFFFF8AE	16
	Timer counter E20	TCNTE20	16	H'FFFFF8B0	16
	Timer counter E21	TCNTE21	16	H'FFFFF8B2	16
	Timer counter E22	TCNTE22	16	H'FFFFF8B4	16
	Timer counter E23	TCNTE23	16	H'FFFFF8B6	16
	Prescaler channel register E20	PSCCRE20	8	H'FFFFF8B8	8
	Prescaler channel register E21	PSCCRE21	8	H'FFFFF8B9	8
	Prescaler channel register E22	PSCCRE22	8	H'FFFFF8BA	8
	Prescaler channel register E23	PSCCRE23	8	H'FFFFF8BB	8
	Timer control register E3	TCRE3	8	H'FFFFF8C0	8
	Timer output control register E3	TOCRE3	8	H'FFFFF8C1	8
	Timer interrupt enable register E3	TIERE3	8	H'FFFFF8C2	8
	Reload control register E3	RLDCRE3	8	H'FFFFF8C3	8
	Timer status register E3	TSRE3	8	H'FFFFF8C4	8
	Prescaler register E3	PSCRE3	8	H'FFFFF8C8	8
	Subblock starting register E3	SSTRE3	8	H'FFFFF8CC	8
	Cycle-setting register E30	CYLRE30	16	H'FFFFF8D0	16
	Cycle-setting register E31	CYLRE31	16	H'FFFFF8D2	16
	Cycle-setting register E32	CYLRE32	16	H'FFFFF8D4	16

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
ATU-III	Cycle-setting register E33	CYLRE33	16	H'FFFFF8D6	16
	Duty cycle setting register E30	DTRE30	16	H'FFFFF8D8	16
	Duty cycle setting register E31	DTRE31	16	H'FFFFF8DA	16
	Duty cycle setting register E32	DTRE32	16	H'FFFFF8DC	16
	Duty cycle setting register E33	DTRE33	16	H'FFFFF8DE	16
	Cycle reload register E30	CRLDE30	16	H'FFFFF8E0	16
	Cycle reload register E31	CRLDE31	16	H'FFFFF8E2	16
	Cycle reload register E32	CRLDE32	16	H'FFFFF8E4	16
	Cycle reload register E33	CRLDE33	16	H'FFFFF8E6	16
	Duty cycle reload register E30	DRLDE30	16	H'FFFFF8E8	16
	Duty cycle reload register E31	DRLDE31	16	H'FFFFF8EA	16
	Duty cycle reload register E32	DRLDE32	16	H'FFFFF8EC	16
	Duty cycle reload register E33	DRLDE33	16	H'FFFFF8EE	16
	Timer counter E30	TCNTE30	16	H'FFFFF8F0	16
	Timer counter E31	TCNTE31	16	H'FFFFF8F2	16
	Timer counter E32	TCNTE32	16	H'FFFFF8F4	16
	Timer counter E33	TCNTE33	16	H'FFFFF8F6	16
	Prescaler channel register E30	PSCCRE30	8	H'FFFFF8F8	8
	Prescaler channel register E31	PSCCRE31	8	H'FFFFF8F9	8
	Prescaler channel register E32	PSCCRE32	8	H'FFFFF8FA	8
	Prescaler channel register E33	PSCCRE33	8	H'FFFFF8FB	8
	Timer control register E4	TCRE4	8	H'FFFFF900	8
	Timer output control register E4	TOCRE4	8	H'FFFFF901	8
	Timer interrupt enable register E4	TIERE4	8	H'FFFFF902	8
	Reload control register E4	RLDCRE4	8	H'FFFFF903	8
	Timer status register E4	TSRE4	8	H'FFFFF904	8
	Prescaler register E4	PSCRE4	8	H'FFFFF908	8
	Subblock starting register E4	SSTRE4	8	H'FFFFF90C	8
	Cycle-setting register E40	CYLRE40	16	H'FFFFF910	16
	Cycle-setting register E41	CYLRE41	16	H'FFFFF912	16

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
ATU-III	Cycle-setting register E42	CYLRE42	16	H'FFFFF914	16
	Cycle-setting register E43	CYLRE43	16	H'FFFFF916	16
	Duty cycle setting register E40	DTRE40	16	H'FFFFF918	16
	Duty cycle setting register E41	DTRE41	16	H'FFFFF91A	16
	Duty cycle setting register E42	DTRE42	16	H'FFFFF91C	16
	Duty cycle setting register E43	DTRE43	16	H'FFFFF91E	16
	Cycle reload register E40	CRLDE40	16	H'FFFFF920	16
	Cycle reload register E41	CRLDE41	16	H'FFFFF922	16
	Cycle reload register E42	CRLDE42	16	H'FFFFF924	16
	Cycle reload register E43	CRLDE43	16	H'FFFFF926	16
	Duty cycle reload register E40	DRLDE40	16	H'FFFFF928	16
	Duty cycle reload register E41	DRLDE41	16	H'FFFFF92A	16
	Duty cycle reload register E42	DRLDE42	16	H'FFFFF92C	16
	Duty cycle reload register E43	DRLDE43	16	H'FFFFF92E	16
	Timer counter E40	TCNTE40	16	H'FFFFF930	16
	Timer counter E41	TCNTE41	16	H'FFFFF932	16
	Timer counter E42	TCNTE42	16	H'FFFFF934	16
	Timer counter E43	TCNTE43	16	H'FFFFF936	16
	Prescaler channel register E40	PSCCRE40	8	H'FFFFF938	8
	Prescaler channel register E41	PSCCRE41	8	H'FFFFF939	8
	Prescaler channel register E42	PSCCRE42	8	H'FFFFF93A	8
	Prescaler channel register E43	PSCCRE43	8	H'FFFFF93B	8
	Timer control register E5	TCRE5	8	H'FFFFF940	8
	Timer output control register E5	TOCRE5	8	H'FFFFF941	8
	Timer interrupt enable register E5	TIERE5	8	H'FFFFF942	8
	Reload control register E5	RLDCRE5	8	H'FFFFF943	8
	Timer status register E5	TSRE5	8	H'FFFFF944	8
	Prescaler register E5	PSCRE5	8	H'FFFFF948	8
	Subblock starting register E5	SSTRE5	8	H'FFFFF94C	8
	Cycle-setting register E50	CYLRE50	16	H'FFFFF950	16
	Cycle-setting register E51	CYLRE51	16	H'FFFFF952	16
	Cycle-setting register E52	CYLRE52	16	H'FFFFF954	16

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
ATU-III	Cycle-setting register E53	CYLRE53	16	H'FFFFF956	16
	Duty cycle setting register E50	DTRE50	16	H'FFFFF958	16
	Duty cycle setting register E51	DTRE51	16	H'FFFFF95A	16
	Duty cycle setting register E52	DTRE52	16	H'FFFFF95C	16
	Duty cycle setting register E53	DTRE53	16	H'FFFFF95E	16
	Cycle reload register E50	CRLDE50	16	H'FFFFF960	16
	Cycle reload register E51	CRLDE51	16	H'FFFFF962	16
	Cycle reload register E52	CRLDE52	16	H'FFFFF964	16
	Cycle reload register E53	CRLDE53	16	H'FFFFF966	16
	Duty cycle reload register E50	DRLDE50	16	H'FFFFF968	16
	Duty cycle reload register E51	DRLDE51	16	H'FFFFF96A	16
	Duty cycle reload register E52	DRLDE52	16	H'FFFFF96C	16
	Duty cycle reload register E53	DRLDE53	16	H'FFFFF96E	16
	Timer counter E50	TCNTE50	16	H'FFFFF970	16
	Timer counter E51	TCNTE51	16	H'FFFFF972	16
	Timer counter E52	TCNTE52	16	H'FFFFF974	16
	Timer counter E53	TCNTE53	16	H'FFFFF976	16
	Prescaler channel register E50	PSCCRE50	8	H'FFFFF978	8
	Prescaler channel register E51	PSCCRE51	8	H'FFFFF979	8
	Prescaler channel register E52	PSCCRE52	8	H'FFFFF97A	8
	Prescaler channel register E53	PSCCRE53	8	H'FFFFF97B	8
	Timer control register E6	TCRE6	8	H'FFFFF980	8
	Timer output control register E6	TOCRE6	8	H'FFFFF981	8
	Timer interrupt enable register E6	TIERE6	8	H'FFFFF982	8
	Reload control register E6	RLDCRE6	8	H'FFFFF983	8
	Timer status register E6	TSRE6	8	H'FFFFF984	8
	Prescaler register E6	PSCRE6	8	H'FFFFF988	8
	Subblock starting register E6	SSTRE6	8	H'FFFFF98C	8

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
ATU-III	Cycle-setting register E60	CYLRE60	16	H'FFFFF990	16
	Cycle-setting register E61	CYLRE61	16	H'FFFFF992	16
	Cycle-setting register E62	CYLRE62	16	H'FFFFF994	16
	Cycle-setting register E63	CYLRE63	16	H'FFFFF996	16
	Duty cycle setting register E60	DTRE60	16	H'FFFFF998	16
	Duty cycle setting register E61	DTRE61	16	H'FFFFF99A	16
	Duty cycle setting register E62	DTRE62	16	H'FFFFF99C	16
	Duty cycle setting register E63	DTRE63	16	H'FFFFF99E	16
	Cycle reload register E60	CRLDE60	16	H'FFFFF9A0	16
	Cycle reload register E61	CRLDE61	16	H'FFFFF9A2	16
	Cycle reload register E62	CRLDE62	16	H'FFFFF9A4	16
	Cycle reload register E63	CRLDE63	16	H'FFFFF9A6	16
	Duty cycle reload register E60	DRLDE60	16	H'FFFFF9A8	16
	Duty cycle reload register E61	DRLDE61	16	H'FFFFF9AA	16
	Duty cycle reload register E62	DRLDE62	16	H'FFFFF9AC	16
	Duty cycle reload register E63	DRLDE63	16	H'FFFFF9AE	16
	Timer counter E60	TCNTE60	16	H'FFFFF9B0	16
	Timer counter E61	TCNTE61	16	H'FFFFF9B2	16
	Timer counter E62	TCNTE62	16	H'FFFFF9B4	16
	Timer counter E63	TCNTE63	16	H'FFFFF9B6	16
	Prescaler channel register E60	PSCCRE60	8	H'FFFFF9B8	8
	Prescaler channel register E61	PSCCRE61	8	H'FFFFF9B9	8
	Prescaler channel register E62	PSCCRE62	8	H'FFFFF9BA	8
	Prescaler channel register E63	PSCCRE63	8	H'FFFFF9BB	8
	Timer start register F	TSTRF	32	H'FFFFFA00	8, 16, 32
	Noise canceller control register F	NCCRF	32	H'FFFFFA04	8, 16, 32
	Noise cancellation mode channel register 1F	NCMCR1F	32	H'FFFFFA0C	32

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
ATU-III	Noise canceler counter FA0	NCNTFA0	8	H'FFFFFA10	8
	Noise cancel register FA0	NCRFA0	8	H'FFFFFA11	8
	Noise canceler counter FA1	NCNTFA1	8	H'FFFFFA12	8
	Noise cancel register FA1	NCRFA1	8	H'FFFFFA13	8
	Noise canceler counter FA2	NCNTFA2	8	H'FFFFFA14	8
	Noise cancel register FA2	NCRFA2	8	H'FFFFFA15	8
	Noise canceler counter FA3	NCNTFA3	8	H'FFFFFA16	8
	Noise cancel register FA3	NCRFA3	8	H'FFFFFA17	8
	Noise canceler counter FA4	NCNTFA4	8	H'FFFFFA18	8
	Noise cancel register FA4	NCRFA4	8	H'FFFFFA19	8
	Noise canceler counter FA5	NCNTFA5	8	H'FFFFFA1A	8
	Noise cancel register FA5	NCRFA5	8	H'FFFFFA1B	8
	Noise canceler counter FA6	NCNTFA6	8	H'FFFFFA1C	8
	Noise cancel register FA6	NCRFA6	8	H'FFFFFA1D	8
	Noise canceler counter FA7	NCNTFA7	8	H'FFFFFA1E	8
	Noise cancel register FA7	NCRFA7	8	H'FFFFFA1F	8
	Noise canceler counter FA8	NCNTFA8	8	H'FFFFFA20	8
	Noise cancel register FA8	NCRFA8	8	H'FFFFFA21	8
	Noise canceler counter FA9	NCNTFA9	8	H'FFFFFA22	8
	Noise cancel register FA9	NCRFA9	8	H'FFFFFA23	8
	Noise canceler counter FA10	NCNTFA10	8	H'FFFFFA24	8
	Noise cancel register FA10	NCRFA10	8	H'FFFFFA25	8
	Noise canceler counter FA11	NCNTFA11	8	H'FFFFFA26	8
	Noise cancel register FA11	NCRFA11	8	H'FFFFFA27	8
	Noise canceler counter FA12	NCNTFA12	8	H'FFFFFA28	8
	Noise cancel register FA12	NCRFA12	8	H'FFFFFA29	8
	Noise canceler counter FA13	NCNTFA13	8	H'FFFFFA2A	8
	Noise cancel register FA13	NCRFA13	8	H'FFFFFA2B	8
	Noise canceler counter FA14	NCNTFA14	8	H'FFFFFA2C	8
	Noise cancel register FA14	NCRFA14	8	H'FFFFFA2D	8

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
ATU-III	Noise canceler counter FA15	NCNTFA15	8	H'FFFFFA2E	8
	Noise cancel register FA15	NCRFA15	8	H'FFFFFA2F	8
	Noise canceler counter FA16	NCNTFA16	8	H'FFFFFA30	8
	Noise cancel register FA16	NCRFA16	8	H'FFFFFA31	8
	Noise canceler counter FA17	NCNTFA17	8	H'FFFFFA32	8
	Noise cancel register FA17	NCRFA17	8	H'FFFFFA33	8
	Noise canceler counter FA18	NCNTFA18	8	H'FFFFFA34	8
	Noise cancel register FA18	NCRFA18	8	H'FFFFFA35	8
	Noise canceler counter FA19	NCNTFA19	8	H'FFFFFA36	8
	Noise cancel register FA19	NCRFA19	8	H'FFFFFA37	8
	Noise canceler counter FA20	NCNTFA20	8	H'FFFFFA38	8
	Noise cancel register FA20	NCRFA20	8	H'FFFFFA39	8
	Noise canceler counter FA21	NCNTFA21	8	H'FFFFFA3A	8
	Noise cancel register FA21	NCRFA21	8	H'FFFFFA3B	8
	Noise canceler counter FA22	NCNTFA22	8	H'FFFFFA3C	8
	Noise cancel register FA22	NCRFA22	8	H'FFFFFA3D	8
	Noise canceler counter FA23	NCNTFA23	8	H'FFFFFA3E	8
	Noise cancel register FA23	NCRFA23	8	H'FFFFFA3F	8
	Noise canceler counter FA24	NCNTFA24	8	H'FFFFFA40	8
	Noise cancel register FA24	NCRFA24	8	H'FFFFFA41	8
	Noise canceler counter FA25	NCNTFA25	8	H'FFFFFA42	8
	Noise cancel register FA25	NCRFA25	8	H'FFFFFA43	8
	Noise canceler counter FA26	NCNTFA26	8	H'FFFFFA44	8
	Noise cancel register FA26	NCRFA26	8	H'FFFFFA45	8
	Noise canceler counter FA27	NCNTFA27	8	H'FFFFFA46	8
	Noise cancel register FA27	NCRFA27	8	H'FFFFFA47	8
	Noise canceler counter FB0	NCNTFB0	8	H'FFFFFA50	8
	Noise cancel register FB0	NCRFB0	8	H'FFFFFA51	8

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
ATU-III	Noise canceler counter FB1	NCNTFB1	8	H'FFFFFA52	8
	Noise cancel register FB1	NCRFB1	8	H'FFFFFA53	8
	Noise canceler counter FB2	NCNTFB2	8	H'FFFFFA54	8
	Noise cancel register FB2	NCRFB2	8	H'FFFFFA55	8
	Timer control register F0	TCRF0	8	H'FFFFFA80	8
	Timer interrupt enable register F0	TIERF0	8	H'FFFFFA81	8
	Timer status register F0	TSRF0	8	H'FFFFFA83	8
	Timer counter AF0	ECNTAF0	32	H'FFFFFA84	32
	Event counter F0	ECNTBF0	16	H'FFFFFA88	16
	General register BF0	GRBF0	16	H'FFFFFA8A	16
	Time counter CF0	ECNTCF0	32	H'FFFFFA8C	32
	General register AF0	GRAF0	32	H'FFFFFA90	32
	Capture output register F0	CDRF0	32	H'FFFFFA94	32
	General register CF0	GRCF0	32	H'FFFFFA98	32
	Timer control register F1	TCRF1	8	H'FFFFFAA0	8
	Timer interrupt enable register F1	TIERF1	8	H'FFFFFAA1	8
	Timer status register F1	TSRF1	8	H'FFFFFAA3	8
	Timer counter AF1	ECNTAF1	32	H'FFFFFAA4	32
	Event counter F1	ECNTBF1	16	H'FFFFFAA8	16
	General register BF1	GRBF1	16	H'FFFFFAAA	16
	Time counter CF1	ECNTCF1	32	H'FFFFFAAC	32
	General register AF1	GRAF1	32	H'FFFFFAB0	32
	Capture output register F1	CDRF1	32	H'FFFFFAB4	32
	General register CF1	GRCF1	32	H'FFFFFAB8	32
	Timer control register F2	TCRF2	8	H'FFFFFAC0	8
	Timer interrupt enable register F2	TIERF2	8	H'FFFFFAC1	8
	Timer status register F2	TSRF2	8	H'FFFFFAC3	8
	Timer counter AF2	ECNTAF2	32	H'FFFFFAC4	32

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
ATU-III	Event counter F2	ECNTBF2	16	H'FFFFFAC8	16
	General register BF2	GRBF2	16	H'FFFFFACA	16
	Time counter CF2	ECNTCF2	32	H'FFFFFACC	32
	General register AF2	GRAF2	32	H'FFFFFAD0	32
	Capture output register F2	CDRF2	32	H'FFFFFAD4	32
	General register CF2	GRCF2	32	H'FFFFFAD8	32
	Timer control register F3	TCRF3	8	H'FFFFFAE0	8
	Timer interrupt enable register F3	TIERF3	8	H'FFFFFAE1	8
	Timer status register F3	TSRF3	8	H'FFFFFAE3	8
	Timer counter AF3	ECNTAF3	32	H'FFFFFAE4	32
	Event counter F3	ECNTBF3	16	H'FFFFFAE8	16
	General register BF3	GRBF3	16	H'FFFFFAEA	16
	Time counter CF3	ECNTCF3	32	H'FFFFFAEC	32
	General register AF3	GRAF3	32	H'FFFFFAF0	32
	Capture output register F3	CDRF3	32	H'FFFFFAF4	32
	General register CF3	GRCF3	32	H'FFFFFAF8	32
	Timer control register F4	TCRF4	8	H'FFFFFB00	8
	Timer interrupt enable register F4	TIERF4	8	H'FFFFFB01	8
	Timer status register F4	TSRF4	8	H'FFFFFB03	8
	Timer counter AF4	ECNTAF4	32	H'FFFFFB04	32
	Event counter F4	ECNTBF4	16	H'FFFFFB08	16
	General register BF4	GRBF4	16	H'FFFFFB0A	16
	Time counter CF4	ECNTCF4	32	H'FFFFFB0C	32
	General register AF4	GRAF4	32	H'FFFFFB10	32
	Capture output register F4	CDRF4	32	H'FFFFFB14	32
	General register CF4	GRCF4	32	H'FFFFFB18	32
	Timer control register F5	TCRF5	8	H'FFFFFB20	8
	Timer interrupt enable register F5	TIERF5	8	H'FFFFFB21	8
	Timer status register F5	TSRF5	8	H'FFFFFB23	8
	Timer counter AF5	ECNTAF5	32	H'FFFFFB24	32

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
ATU-III	Event counter F5	ECNTBF5	16	H'FFFFFB28	16
	General register BF5	GRBF5	16	H'FFFFFB2A	16
	Time counter CF5	ECNTCF5	32	H'FFFFFB2C	32
	General register AF5	GRAF5	32	H'FFFFFB30	32
	Capture output register F5	CDRF5	32	H'FFFFFB34	32
	General register CF5	GRCF5	32	H'FFFFFB38	32
	Timer control register F6	TCRF6	8	H'FFFFFB40	8
	Timer interrupt enable register F6	TIERF6	8	H'FFFFFB41	8
	Timer status register F6	TSRF6	8	H'FFFFFB43	8
	Timer counter AF6	ECNTAF6	32	H'FFFFFB44	32
	Event counter F6	ECNTBF6	16	H'FFFFFB48	16
	General register BF6	GRBF6	16	H'FFFFFB4A	16
	Time counter CF6	ECNTCF6	32	H'FFFFFB4C	32
	General register AF6	GRAF6	32	H'FFFFFB50	32
	Capture output register F6	CDRF6	32	H'FFFFFB54	32
	General register CF6	GRCF6	32	H'FFFFFB58	32
	Timer control register F7	TCRF7	8	H'FFFFFB60	8
	Timer interrupt enable register F7	TIERF7	8	H'FFFFFB61	8
	Timer status register F7	TSRF7	8	H'FFFFFB63	8
	Timer counter AF7	ECNTAF7	32	H'FFFFFB64	32
	Event counter F7	ECNTBF7	16	H'FFFFFB68	16
	General register BF7	GRBF7	16	H'FFFFFB6A	16
	Time counter CF7	ECNTCF7	32	H'FFFFFB6C	32
	General register AF7	GRAF7	32	H'FFFFFB70	32
	Capture output register F7	CDRF7	32	H'FFFFFB74	32
	General register CF7	GRCF7	32	H'FFFFFB78	32
	Timer control register F8	TCRF8	8	H'FFFFFB80	8
	Timer interrupt enable register F8	TIERF8	8	H'FFFFFB81	8
	Timer status register F8	TSRF8	8	H'FFFFFB83	8
	Timer counter AF8	ECNTAF8	32	H'FFFFFB84	32
	Event counter F8	ECNTBF8	16	H'FFFFFB88	16

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
ATU-III	General register BF8	GRBF8	16	H'FFFFFB8A	16
	Time counter CF8	ECNTCF8	32	H'FFFFFB8C	32
	General register AF8	GRAF8	32	H'FFFFFB90	32
	Capture output register F8	CDRF8	32	H'FFFFFB94	32
	General register CF8	GRCF8	32	H'FFFFFB98	32
	Timer control register F9	TCRF9	8	H'FFFFFBA0	8
	Timer interrupt enable register F9	TIERF9	8	H'FFFFFBA1	8
	Timer status register F9	TSRF9	8	H'FFFFFBA3	8
	Timer counter AF9	ECNTAF9	32	H'FFFFFBA4	32
	Event counter F9	ECNTBF9	16	H'FFFFFBA8	16
	General register BF9	GRBF9	16	H'FFFFFBAA	16
	Time counter CF9	ECNTCF9	32	H'FFFFFBAC	32
	General register AF9	GRAF9	32	H'FFFFFBB0	32
	Capture output register F9	CDRF9	32	H'FFFFFBB4	32
	General register CF9	GRCF9	32	H'FFFFFBB8	32
	Timer control register F10	TCRF10	8	H'FFFFFBC0	8
	Timer interrupt enable register F10	TIERF10	8	H'FFFFFBC1	8
	Timer status register F10	TSRF10	8	H'FFFFFBC3	8
	Timer counter AF10	ECNTAF10	32	H'FFFFFBC4	32
	Event counter F10	ECNTBF10	16	H'FFFFFBC8	16
	General register BF10	GRBF10	16	H'FFFFFBCA	16
	Time counter CF10	ECNTCF10	32	H'FFFFFBCC	32
	General register AF10	GRAF10	32	H'FFFFFBD0	32
	Capture output register F10	CDRF10	32	H'FFFFFBD4	32
	General register CF10	GRCF10	32	H'FFFFFBD8	32
	Timer control register F11	TCRF11	8	H'FFFFFBE0	8
	Timer interrupt enable register F11	TIERF11	8	H'FFFFFBE1	8
	Timer status register F11	TSRF11	8	H'FFFFFBE3	8
	Timer counter AF11	ECNTAF11	32	H'FFFFFBE4	32

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
ATU-III	Event counter F11	ECNTBF11	16	H'FFFFFBE8	16
	General register BF11	GRBF11	16	H'FFFFFBEA	16
	Time counter CF11	ECNTCF11	32	H'FFFFFBEC	32
	General register AF11	GRAF11	32	H'FFFFFBF0	32
	Capture output register F11	CDRF11	32	H'FFFFFBF4	32
	General register CF11	GRCF11	32	H'FFFFFBF8	32
	Timer control register F12	TCRF12	8	H'FFFFFC00	8
	Timer interrupt enable register F125	TIERF12	8	H'FFFFFC01	8
	Timer status register F12	TSRF12	8	H'FFFFFC03	8
	Timer counter AF12	ECNTAF12	32	H'FFFFFC04	32
	Event counter F12	ECNTBF12	16	H'FFFFFC08	16
	General register BF12	GRBF12	16	H'FFFFFC0A	16
	Time counter CF12	ECNTCF12	32	H'FFFFFC0C	32
	General register AF12	GRAF12	32	H'FFFFFC10	32
	Capture output register F12	CDRF12	32	H'FFFFFC14	32
	General register CF12	GRCF12	32	H'FFFFFC18	32
	General register DF12	GRDF12	32	H'FFFFFC1C	32
	Timer control register F13	TCRF13	8	H'FFFFFC20	8
	Timer interrupt enable register F13	TIERF13	8	H'FFFFFC21	8
	Timer status register F13	TSRF13	8	H'FFFFFC23	8
	Timer counter AF13	ECNTAF13	32	H'FFFFFC24	32
	Event counter F13	ECNTBF13	16	H'FFFFFC28	16
	General register BF13	GRBF13	16	H'FFFFFC2A	16
	Timer counter CF13	ECNTCF13	32	H'FFFFFC2C	32
	General register AF13	GRAF13	32	H'FFFFFC30	32
	Capture output register F13	CDRF13	32	H'FFFFFC34	32
	General register CF13	GRCF13	32	H'FFFFFC38	32
	General register DF13	GRDF13	32	H'FFFFFC3C	32
	Timer control register F14	TCRF14	8	H'FFFFFC40	8

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
ATU-III	Timer interrupt enable register F14	TIERF14	8	H'FFFFFC41	8
	Timer status register F14	TSRF14	8	H'FFFFFC43	8
	Timer counter AF14	ECNTAF14	32	H'FFFFFC44	32
	Event counter F14	ECNTBF14	16	H'FFFFFC48	16
	General register BF14	GRBF14	16	H'FFFFFC4A	16
	Time counter CF14	ECNTCF14	32	H'FFFFFC4C	32
	General register AF14	GRAF14	32	H'FFFFFC50	32
	Capture output register F14	CDRF14	32	H'FFFFFC54	32
	General register CF14	GRCF14	32	H'FFFFFC58	32
	General register DF14	GRDF14	32	H'FFFFFC5C	32
	Timer control register F15	TCRF15	8	H'FFFFFC60	8
	Timer interrupt enable register F15	TIERF15	8	H'FFFFFC61	8
	Timer status register F15	TSRF15	8	H'FFFFFC63	8
	Timer counter AF15	ECNTAF15	32	H'FFFFFC64	32
	Event counter F15	ECNTBF15	16	H'FFFFFC68	16
	General register BF15	GRBF15	16	H'FFFFFC6A	16
	Time counter CF15	ECNTCF15	32	H'FFFFFC6C	32
	General register AF15	GRAF15	32	H'FFFFFC70	32
	Capture output register F15	CDRF15	32	H'FFFFFC74	32
	General register CF15	GRCF15	32	H'FFFFFC78	32
	General register DF15	GRDF15	32	H'FFFFFC7C	32
	Timer control register F16	TCRF16	8	H'FFFFFC80	8
	Timer interrupt enable register F16	TIERF16	8	H'FFFFFC81	8
	Timer status register F16	TSRF16	8	H'FFFFFC83	8
	Timer counter AF16	ECNTAF16	32	H'FFFFFC84	32
	Event counter F16	ECNTBF16	16	H'FFFFFC88	16
	General register BF16	GRBF16	16	H'FFFFFC8A	16
	Time counter CF16	ECNTCF16	32	H'FFFFFC8C	32
	General register AF16	GRAF16	32	H'FFFFFC90	32
	Capture output register F16	CDRF16	32	H'FFFFFC94	32

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
ATU-III	General register CF16	GRCF16	32	H'FFFFFC98	32
	Timer control register F17	TCRF17	8	H'FFFFFCA0	8
	Timer interrupt enable register F17	TIERF17	8	H'FFFFFCA1	8
	Timer status register F17	TSRF17	8	H'FFFFFCA3	8
	Timer counter AF17	ECNTAF17	32	H'FFFFFCA4	32
	Event counter F17	ECNTBF17	16	H'FFFFFCA8	16
	General register BF17	GRBF17	16	H'FFFFFCAA	16
	Time counter CF17	ECNTCF17	32	H'FFFFFCAC	32
	General register AF17	GRAF17	32	H'FFFFFCB0	32
	Capture output register F17	CDRF17	32	H'FFFFFCB4	32
	General register CF17	GRCF17	32	H'FFFFFCB8	32
	Timer control register F18	TCRF18	8	H'FFFFFCC0	8
	Timer interrupt enable register F18	TIERF18	8	H'FFFFFCC1	8
	Timer status register F18	TSRF18	8	H'FFFFFCC3	8
	Timer counter AF18	ECNTAF18	32	H'FFFFFCC4	32
	Event counter F18	ECNTBF18	16	H'FFFFFCC8	16
	General register BF18	GRBF18	16	H'FFFFFCCA	16
	Time counter CF18	ECNTCF18	32	H'FFFFFCCC	32
	General register AF18	GRAF18	32	H'FFFFFCD0	32
	Capture output register F18	CDRF18	32	H'FFFFFCD4	32
	General register CF18	GRCF18	32	H'FFFFFCD8	32
	Timer control register F19	TCRF19	8	H'FFFFFCE0	8
	Timer interrupt enable register F19	TIERF19	8	H'FFFFFCE1	8
	Timer status register F19	TSRF19	8	H'FFFFFCE3	8
	Timer counter AF19	ECNTAF19	32	H'FFFFFCE4	32
	Event counter F19	ECNTBF19	16	H'FFFFFCE8	16
	General register BF19	GRBF19	16	H'FFFFFCEA	16
	Time counter CF19	ECNTCF19	32	H'FFFFFCEC	32
	General register AF19	GRAF19	32	H'FFFFFCF0	32
	Capture output register F19	CDRF19	32	H'FFFFFCF4	32

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
ATU-III	General register CF19	GRCF19	32	H'FFFFFCF8	32
	Timer control register F20	TCRF20	8	H'FFFFFD00	8
	Timer interrupt enable register F20	TIERF20	8	H'FFFFFD01	8
	Timer status register F20	TSRF20	8	H'FFFFFD03	8
	Timer counter AF20	ECNTAF20	32	H'FFFFFD04	32
	Event counter F20	ECNTBF20	16	H'FFFFFD08	16
	General register BF20	GRBF20	16	H'FFFFFD0A	16
	Time counter CF20	ECNTCF20	32	H'FFFFFD0C	32
	General register AF20	GRAF20	32	H'FFFFFD10	32
	Capture output register F20	CDRF20	32	H'FFFFFD14	32
	General register CF20	GRCF20	32	H'FFFFFD18	32
	Timer control register F21	TCRF21	8	H'FFFFFD20	8
	Timer interrupt enable register F21	TIERF21	8	H'FFFFFD21	8
	Timer status register F21	TSRF21	8	H'FFFFFD23	8
	Timer counter AF21	ECNTAF21	32	H'FFFFFD24	32
	Event counter F21	ECNTBF21	16	H'FFFFFD28	16
	General register BF21	GRBF21	16	H'FFFFFD2A	16
	Time counter CF21	ECNTCF21	32	H'FFFFFD2C	32
	General register AF21	GRAF21	32	H'FFFFFD30	32
	Capture output register F21	CDRF21	32	H'FFFFFD34	32
	General register CF21	GRCF21	32	H'FFFFFD38	32
	Timer control register F22	TCRF22	8	H'FFFFFD40	8
	Timer interrupt enable register F22	TIERF22	8	H'FFFFFD41	8
	Timer status register F22	TSRF22	8	H'FFFFFD43	8
	Timer counter AF22	ECNTAF22	32	H'FFFFFD44	32
	Event counter F22	ECNTBF22	16	H'FFFFFD48	16
	General register BF22	GRBF22	16	H'FFFFFD4A	16
	Time counter CF22	ECNTCF22	32	H'FFFFFD4C	32
	General register AF22	GRAF22	32	H'FFFFFD50	32

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
ATU-III	Capture output register F22	CDRF22	32	H'FFFFFD54	32
	General register CF22	GRCF22	32	H'FFFFFD58	32
	Timer control register F23	TCRF23	8	H'FFFFFD60	8
	Timer interrupt enable register F23	TIERF23	8	H'FFFFFD61	8
	Timer status register F23	TSRF23	8	H'FFFFFD63	8
	Timer counter AF23	ECNTAF23	32	H'FFFFFD64	32
	Event counter F23	ECNTBF23	16	H'FFFFFD68	16
	General register BF23	GRBF23	16	H'FFFFFD6A	16
	Time counter CF23	ECNTCF23	32	H'FFFFFD6C	32
	General register AF23	GRAF23	32	H'FFFFFD70	32
	Capture output register F23	CDRF23	32	H'FFFFFD74	32
	General register CF23	GRCF23	32	H'FFFFFD78	32
	Timer control register F24	TCRF24	8	H'FFFFFD80	8
	Timer interrupt enable register F24	TIERF24	8	H'FFFFFD81	8
	Timer status register F24	TSRF24	8	H'FFFFFD83	8
	Timer counter AF24	ECNTAF24	32	H'FFFFFD84	32
	Event counter F24	ECNTBF24	16	H'FFFFFD88	16
	General register BF24	GRBF24	16	H'FFFFFD8A	16
	Time counter CF24	ECNTCF24	32	H'FFFFFD8C	32
	General register AF24	GRAF24	32	H'FFFFFD90	32
	Capture output register F24	CDRF24	32	H'FFFFFD94	32
	General register CF24	GRCF24	32	H'FFFFFD98	32
	Timer control register F25	TCRF25	8	H'FFFFFDA0	8
	Timer interrupt enable register F25	TIERF25	8	H'FFFFFDA1	8
	Timer status register F25	TSRF25	8	H'FFFFFDA3	8
	Timer counter AF25	ECNTAF25	32	H'FFFFFDA4	32
	Event counter F25	ECNTBF25	16	H'FFFFFDA8	16
	General register BF25	GRBF25	16	H'FFFFFDAA	16

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
ATU-III	Time counter CF25	ECNTCF25	32	H'FFFFFDAC	32
	General register AF25	GRAF25	32	H'FFFFFDB0	32
	Capture output register F25	CDRF25	32	H'FFFFFDB4	32
	General register CF25	GRCF25	32	H'FFFFFDB8	32
	Timer control register F26	TCRF26	8	H'FFFFFDC0	8
	Timer interrupt enable register F26	TIERF26	8	H'FFFFFDC1	8
	Timer status register F26	TSRF26	8	H'FFFFFDC3	8
	Timer counter AF26	ECNTAF26	32	H'FFFFFDC4	32
	Event counter F26	ECNTBF26	16	H'FFFFFDC8	16
	General register BF26	GRBF26	16	H'FFFFFDCA	16
	Time counter CF26	ECNTCF26	32	H'FFFFFDCC	32
	General register AF26	GRAF26	32	H'FFFFFDD0	32
	Capture output register F26	CDRF26	32	H'FFFFFDD4	32
	General register CF26	GRCF26	32	H'FFFFFDD8	32
	Timer control register F27	TCRF27	8	H'FFFFFDE0	8
	Timer interrupt enable register F27	TIERF27	8	H'FFFFFDE1	8
	Timer status register F27	TSRF27	8	H'FFFFFDE3	8
	Timer counter AF27	ECNTAF27	32	H'FFFFFDE4	32
	Event counter F27	ECNTBF27	16	H'FFFFFDE8	16
	General register BF27	GRBF27	16	H'FFFFFDEA	16
	Time counter CF27	ECNTCF27	32	H'FFFFFDEC	32
	General register AF27	GRAF27	32	H'FFFFFDF0	32
	Capture output register F27	CDRF27	32	H'FFFFFDF4	32
	General register CF27	GRCF27	32	H'FFFFFDF8	32
	Timer start register G	TSTRG	8	H'FFFFFE01	8
	Timer control register G0	TCRG0	8	H'FFFFFE80	8
	Timer status register G0	TSRG0	8	H'FFFFFE81	8

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
ATU-III	Timer counter G0	TCNTG0	16	H'FFFFFE84	16
	Compare match register G0	OCRG0	16	H'FFFFFE86	16
	Timer control register G1	TCRG1	8	H'FFFFFE90	8
	Timer status register G1	TSRG1	8	H'FFFFFE91	8
	Timer counter G1	TCNTG1	16	H'FFFFFE94	16
	Compare match register G1	OCRG1	16	H'FFFFFE96	16
	Timer control register G2	TCRG2	8	H'FFFFFEA0	8
	Timer status register G2	TSRG2	8	H'FFFFFEA1	8
	Timer counter G2	TCNTG2	16	H'FFFFFEA4	16
	Compare match register G2	OCRG2	16	H'FFFFFEA6	16
	Timer control register G3	TCRG3	8	H'FFFFFEB0	8
	Timer status register G3	TSRG3	8	H'FFFFFEB1	8
	Timer counter G3	TCNTG3	16	H'FFFFFEB4	16
	Compare match register G3	OCRG3	16	H'FFFFFEB6	16
	Timer control register G4	TCRG4	8	H'FFFFFEC0	8
	Timer status register G4	TSRG4	8	H'FFFFFEC1	8
	Timer counter G4	TCNTG4	16	H'FFFFFEC4	16
	Compare match register G4	OCRG4	16	H'FFFFFEC6	16
	Timer control register G5	TCRG5	8	H'FFFFFED0	8
	Timer status register G5	TSRG5	8	H'FFFFFED1	8
	Timer counter G5	TCNTG5	16	H'FFFFFED4	16
	Compare match register G5	OCRG5	16	H'FFFFFED6	16
	Timer control register H	TCRH	8	H'FFFFFF40	8
	Timer status register H	TSRH	8	H'FFFFFF41	8
	Timer counter 1H	TCNT1H	16	H'FFFFFF44	16
	Compare match register 1H	OCR1H	16	H'FFFFFF46	16
	Timer counter 2H	TCNT2H	32	H'FFFFFF48	32
	Timer start register J	TSTRJ	8	H'FFFFFF80	8

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
ATU-III	Timer control register J0	TCRJ0	8	H'FFFFFF90	8
	FIFO control register J0	FCRJ0	8	H'FFFFFF91	8
	Timer status register J0	TSRJ0	8	H'FFFFFF92	8
	Timer interrupt enable register J0	TIERJ0	8	H'FFFFFF94	8
	FIFO data count register J0	FDNRJ0	8	H'FFFFFF95	8
	Noise canceler counter J0	NCNTJ0	8	H'FFFFFF96	8
	Noise cancel register J0	NCRJ0	8	H'FFFFFF97	8
	Timer counter J0	TCNTJ0	16	H'FFFFFF98	16
	Compare match register J0	OCRJ0	16	H'FFFFFF9A	16
	FIFO register J0	FIFOJ0	16	H'FFFFFF9C	16
	Timer control register J1	TCRJ1	8	H'FFFFFFA0	8
	FIFO control register J1	FCRJ1	8	H'FFFFFFA1	8
	Timer status register J1	TSRJ1	8	H'FFFFFFA2	8
	Timer interrupt enable register J1	TIERJ1	8	H'FFFFFFA4	8
	FIFO data count register J1	FDNRJ1	8	H'FFFFFFA5	8
	Noise canceler counter J1	NCNTJ1	8	H'FFFFFFA6	8
	Noise cancel register J1	NCRJ1	8	H'FFFFFFA7	8
	Timer counter J1	TCNTJ1	16	H'FFFFFFA8	16
	Compare match register J1	OCRJ1	16	H'FFFFFFAA	16
	FIFO register J1	FIFOJ1	16	H'FFFFFFAC	16
WDT	Watchdog timer control register	WTCR	16	H'FFFE0000	8, 16
	Watchdog timer counter	WTCNT	16	H'FFFE0002	8, 16
	Watchdog timer status register	WTSR	16	H'FFFE0004	8, 16
	Watchdog reset control register	WRCR	16	H'FFFE0006	8, 16
CMT	Compare match timer start register	CMSTR	16	H'FFFEC000	16
	Compare match timer control register_0	CMCR_0	8	H'FFFEC010	8

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
CMT	Compare match timer status register_0	CMSR_0	8	H'FFFEC011	8
	Compare match counter_0	CMCNT_0	16	H'FFFEC012	16
	Compare match constant register_0	CMCOR_0	16	H'FFFEC014	16
	Compare match timer control register_1	CMCR_1	8	H'FFFEC020	8
	Compare match timer status register_1	CMSR_1	8	H'FFFEC021	8
	Compare match counter_1	CMCNT_1	16	H'FFFEC022	16
	Compare match constant register_1	CMCOR_1	16	H'FFFEC024	16
SCI	Serial mode register	SCSMR1A	8	H'FFFF8000	8
	Bit rate register	SCBRR1A	8	H'FFFF8004	8
	Serial control register	SCSCR1A	8	H'FFFF8008	8
	Transmit data register	SCTDR1A	8	H'FFFF800C	8
	Serial status register	SCSSR1A	8	H'FFFF8010	8
	Receive data register	SCRDR1A	8	H'FFFF8014	8
	Serial mode register	SCSMR1B	8	H'FFFF8800	8
	Bit rate register	SCBRR1B	8	H'FFFF8804	8
	Serial control register	SCSCR1B	8	H'FFFF8808	8
	Transmit data register	SCTDR1B	8	H'FFFF880C	8
	Serial status register	SCSSR1B	8	H'FFFF8810	8
	Receive data register	SCRDR1B	8	H'FFFF8814	8
	Serial mode register	SCSMR1C	8	H'FFFF9000	8
	Bit rate register	SCBRR1C	8	H'FFFF9004	8
	Serial control register	SCSCR1C	8	H'FFFF9008	8
	Transmit data register	SCTDR1C	8	H'FFFF900C	8
	Serial status register	SCSSR1C	8	H'FFFF9010	8
	Receive data register	SCRDR1C	8	H'FFFF9014	8
	Serial mode register	SCSMR1D	8	H'FFFF9800	8

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
SCI	Bit rate register	SCBRR1D	8	H'FFFF9804	8
	Serial control register	SCSCR1D	8	H'FFFF9808	8
	Transmit data register	SCTDR1D	8	H'FFFF980C	8
	Serial status register	SCSSR1D	8	H'FFFF9810	8
	Receive data register	SCRDR1D	8	H'FFFF9814	8
	Serial mode register	SCSMR1E	8	H'FFFFA000	8
	Bit rate register	SCBRR1E	8	H'FFFFA004	8
	Serial control register	SCSCR1E	8	H'FFFFA008	8
	Transmit data register	SCTDR1E	8	H'FFFFA00C	8
	Serial status register	SCSSR1E	8	H'FFFFA010	8
	Receive data register	SCRDR1E	8	H'FFFFA014	8
RSPI	RSPI control register A	SPCRA	8	H'FFFFB000	8, 16
	RSPI slave select polarity register A	SSLPA	8	H'FFFFB001	8
	RSPI pin control register A	SPPCRA	8	H'FFFFB002	8, 16
	RSPI status register A	SPSRA	8	H'FFFFB003	8
	RSPI data register A	SPDRA	16	H'FFFFB004	16
	RSPI sequence control register A	SPSCRA	8	H'FFFFB008	8, 16
	RSPI sequence status register A	SPSSRA	8	H'FFFFB009	8
	RSPI bit rate register A	SPBRA	8	H'FFFFB00A	8
	RSPI clock delay register A	SPCKDA	8	H'FFFFB00C	8, 16
	RSPI slave select negate delay register A	SSLNDA	8	H'FFFFB00D	8
	RSPI next-access delay register A	SPNDA	8	H'FFFFB00E	8
	RSPI command register A0	SPCMDA0	16	H'FFFFB010	16
	RSPI command register A1	SPCMDA1	16	H'FFFFB012	16
	RSPI command register A2	SPCMDA2	16	H'FFFFB014	16
	RSPI command register A3	SPCMDA3	16	H'FFFFB016	16
	RSPI command register A4	SPCMDA4	16	H'FFFFB018	16
	RSPI command register A5	SPCMDA5	16	H'FFFFB01A	16

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
RSPI	RSPI command register A6	SPCMDA6	16	H'FFFFB01C	16
	RSPI command register A7	SPCMDA7	16	H'FFFFB01E	16
	RSPI control register B	SPCRB	8	H'FFFFB800	8, 16
	RSPI slave select polarity register B	SSLPB	8	H'FFFFB801	8
	RSPI pin control register B	SPPCRB	8	H'FFFFB802	8, 16
	RSPI status register B	SPSRB	8	H'FFFFB803	8
	RSPI data register B	SPDRB	16	H'FFFFB804	16
	RSPI sequence control register B	SPSCRB	8	H'FFFFB808	8, 16
	RSPI sequence status register B	SPSSRB	8	H'FFFFB809	8
	RSPI bit rate register B	SPBRB	8	H'FFFFB80A	8
	RSPI clock delay register B	SPCKDB	8	H'FFFFB80C	8, 16
	RSPI slave select negate delay register B	SSLNDB	8	H'FFFFB80D	8
	RSPI next-access delay register B	SPNDB	8	H'FFFFB80E	8
	RSPI command register B0	SPCMDB0	16	H'FFFFB810	16
	RSPI command register B1	SPCMDB1	16	H'FFFFB812	16
	RSPI command register B2	SPCMDB2	16	H'FFFFB814	16
	RSPI command register B3	SPCMDB3	16	H'FFFFB816	16
	RSPI command register B4	SPCMDB4	16	H'FFFFB818	16
	RSPI command register B5	SPCMDB5	16	H'FFFFB81A	16
	RSPI command register B6	SPCMDB6	16	H'FFFFB81C	16
	RSPI command register B7	SPCMDB7	16	H'FFFFB81E	16
	RSPI control register C	SPCRC	8	H'FFFFC000	8, 16
	RSPI slave select polarity register C	SSLPC	8	H'FFFFC001	8
	RSPI pin control register C	SPPCRC	8	H'FFFFC002	8, 16
	RSPI status register C	SPSRC	8	H'FFFFC003	8
	RSPI data register C	SPDRC	16	H'FFFFC004	16
	RSPI sequence control register C	SPSCRC	8	H'FFFFC008	8, 16

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
RSPI	RSPI sequence status register C	SPSSRC	8	H'FFFFC009	8
	RSPI bit rate register C	SPBRC	8	H'FFFFC00A	8
	RSPI clock delay register C	SPCKDC	8	H'FFFFC00C	8, 16
	RSPI slave select negate delay register C	SSLNDC	8	H'FFFFC00D	8
	RSPI next-access delay register C	SPNDC	8	H'FFFFC00E	8
	RSPI command register C0	SPCMDC0	16	H'FFFFC010	16
	RSPI command register C1	SPCMDC1	16	H'FFFFC012	16
	RSPI command register C2	SPCMDC2	16	H'FFFFC014	16
	RSPI command register C3	SPCMDC3	16	H'FFFFC016	16
	RSPI command register C4	SPCMDC4	16	H'FFFFC018	16
	RSPI command register C5	SPCMDC5	16	H'FFFFC01A	16
	RSPI command register C6	SPCMDC6	16	H'FFFFC01C	16
	RSPI command register C7	SPCMDC7	16	H'FFFFC01E	16
RCAN-TL1 (RCAN_A)	Master control register	MCR	16	H'FFFFD000	16
	General status register	GSR	16	H'FFFFD002	16
	Bit configuration register 1	BCR1	16	H'FFFFD004	16
	Bit configuration register 0	BCR0	16	H'FFFFD006	16
	Interrupt request register	IRR	16	H'FFFFD008	16
	Interrupt mask register	IMR	16	H'FFFFD00A	16
	Error counter register	TEC/REC	16	H'FFFFD00C	16
	Transmit pending register 1	TXPR1	16	H'FFFFD020	32
	Transmit pending register 0	TXPR0	16		
	Transmit cancel register 1	TXCR1	16	H'FFFFD028	16
	Transmit cancel register 0	TXCR0	16	H'FFFFD02A	16
	Transmit acknowledge register 1	TXACK1	16	H'FFFFD030	16
	Transmit acknowledge register 0	TXACK0	16	H'FFFFD032	16
	Abort acknowledge register 1	ABACK1	16	H'FFFFD038	16
	Abort acknowledge register 0	ABACK0	16	H'FFFFD03A	16

Module	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1 (RCAN_A)	Data frame receive pending register 1		RXPR1	16	H'FFFFD040	16
	Data frame receive pending register 0		RXPR0	16	H'FFFFD042	16
	Remote frame receive pending register 1		RFPR1	16	H'FFFFD048	16
	Remote frame receive pending register 0		RFPR0	16	H'FFFFD04A	16
	Mailbox interrupt mask register 1		MBIMR1	16	H'FFFFD050	16
	Mailbox interrupt mask register 0		MBIMR0	16	H'FFFFD052	16
	Unread message status register 1		UMSR1	16	H'FFFFD058	16
	Unread message status register 0		UMSR0	16	H'FFFFD05A	16
	Timer trigger control register 0		TTCR0	16	H'FFFFD080	16
	Cycle maximum/tx-enable window register		CMAX_TEW	16	H'FFFFD084	16
	Reference trigger offset register		RFTROFF	16	H'FFFFD086	16
	Timer status register		TSR	16	H'FFFFD088	16
	Cycle counter register		CCR	16	H'FFFFD08A	16
	Timer counter register		TCNTR	16	H'FFFFD08C	16
	Cycle time register		CYCTR	16	H'FFFFD090	16
	Reference mark register		RFMK	16	H'FFFFD094	16
	Timer compare match register 0		TCMR0	16	H'FFFFD098	16
	Timer compare match register 1		TCMR1	16	H'FFFFD09C	16
	Timer compare match register 2		TCMR2	16	H'FFFFD0A0	16
	Tx-trigger time selection register		TTTSEL	16	H'FFFFD0A4	16
	MB[0].	CONTROL0_H	—	16	H'FFFFD100	16, 32
		CONTROL0_L	—	16	H'FFFFD102	16
		LAFM0	—	16	H'FFFFD104	16, 32
		LAFM1	—	16	H'FFFFD106	16

Module	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1 (RCAN_A)	MB[0].	DATA_01	—	16	H'FFFFD108	8, 16, 32
		DATA_23	—	16	H'FFFFD10A	8, 16
		DATA_45	—	16	H'FFFFD10C	8, 16, 32
		DATA_67	—	16	H'FFFFD10E	8, 16
		CONTROL1	—	16	H'FFFFD110	8, 16
		TIMESTAMP	—	16	H'FFFFD112	16
	MB[1].	CONTROL0_H	—	16	H'FFFFD120	16, 32
		CONTROL0_L	—	16	H'FFFFD122	16
		LAFM0	—	16	H'FFFFD124	16, 32
		LAFM1	—	16	H'FFFFD126	16
		DATA_01	—	16	H'FFFFD128	8, 16, 32
		DATA_23	—	16	H'FFFFD12A	8, 16
		DATA_45	—	16	H'FFFFD12C	8, 16, 32
		DATA_67	—	16	H'FFFFD12E	8, 16
		CONTROL1	—	16	H'FFFFD130	8, 16
		TIMESTAMP	—	16	H'FFFFD132	16
	MB[2].	CONTROL0_H	—	16	H'FFFFD140	16, 32
		CONTROL0_L	—	16	H'FFFFD142	16
		LAFM0	—	16	H'FFFFD144	16, 32
		LAFM1	—	16	H'FFFFD146	16
		DATA_01	—	16	H'FFFFD148	8, 16, 32
		DATA_23	—	16	H'FFFFD14A	8, 16
		DATA_45	—	16	H'FFFFD14C	8, 16, 32
		DATA_67	—	16	H'FFFFD14E	8, 16
		CONTROL1	—	16	H'FFFFD150	8, 16
		TIMESTAMP	—	16	H'FFFFD152	16
	MB[3].	CONTROL0_H	—	16	H'FFFFD160	16, 32
		CONTROL0_L	—	16	H'FFFFD162	16
		LAFM0	—	16	H'FFFFD164	16, 32
		LAFM1	—	16	H'FFFFD166	16

Module	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1 (RCAN_A)	MB[3].	DATA_01	—	16	H'FFFFD168	8, 16, 32
		DATA_23	—	16	H'FFFFD16A	8, 16
		DATA_45	—	16	H'FFFFD16C	8, 16, 32
		DATA_67	—	16	H'FFFFD16E	8, 16
		CONTROL1	—	16	H'FFFFD170	8, 16
		TIMESTAMP	—	16	H'FFFFD172	16
	MB[4].	CONTROL0_H	—	16	H'FFFFD180	16, 32
		CONTROL0_L	—	16	H'FFFFD182	16
		LAFM0	—	16	H'FFFFD184	16, 32
		LAFM1	—	16	H'FFFFD186	16
		DATA_01	—	16	H'FFFFD188	8, 16, 32
		DATA_23	—	16	H'FFFFD18A	8, 16
		DATA_45	—	16	H'FFFFD18C	8, 16, 32
		DATA_67	—	16	H'FFFFD18E	8, 16
		CONTROL1	—	16	H'FFFFD190	8, 16
		TIMESTAMP	—	16	H'FFFFD192	16
	MB[5].	CONTROL0_H	—	16	H'FFFFD1A0	16, 32
		CONTROL0_L	—	16	H'FFFFD1A2	16
		LAFM0	—	16	H'FFFFD1A4	16, 32
		LAFM1	—	16	H'FFFFD1A6	16
		DATA_01	—	16	H'FFFFD1A8	8, 16, 32
		DATA_23	—	16	H'FFFFD1AA	8, 16
		DATA_45	—	16	H'FFFFD1AC	8, 16, 32
		DATA_67	—	16	H'FFFFD1AE	8, 16
		CONTROL1	—	16	H'FFFFD1B0	8, 16
		TIMESTAMP	—	16	H'FFFFD1B2	16
	MB[6].	CONTROL0_H	—	16	H'FFFFD1C0	16, 32
		CONTROL0_L	—	16	H'FFFFD1C2	16
		LAFM0	—	16	H'FFFFD1C4	16, 32
		LAFM1	—	16	H'FFFFD1C6	16

Module	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1 (RCAN_A)	MB[6].	DATA_01	—	16	H'FFFFD1C8	8, 16, 32
		DATA_23	—	16	H'FFFFD1CA	8, 16
		DATA_45	—	16	H'FFFFD1CC	8, 16, 32
		DATA_67	—	16	H'FFFFD1CE	8, 16
		CONTROL1	—	16	H'FFFFD1D0	8, 16
		TIMESTAMP	—	16	H'FFFFD1D2	16
	MB[7].	CONTROL0_H	—	16	H'FFFFD1E0	16, 32
		CONTROL0_L	—	16	H'FFFFD1E2	16
		LAFM0	—	16	H'FFFFD1E4	16, 32
		LAFM1	—	16	H'FFFFD1E6	16
		DATA_01	—	16	H'FFFFD1E8	8, 16, 32
		DATA_23	—	16	H'FFFFD1EA	8, 16
		DATA_45	—	16	H'FFFFD1EC	8, 16, 32
		DATA_67	—	16	H'FFFFD1EE	8, 16
		CONTROL1	—	16	H'FFFFD1F0	8, 16
		TIMESTAMP	—	16	H'FFFFD1F2	16
	MB[8].	CONTROL0_H	—	16	H'FFFFD200	16, 32
		CONTROL0_L	—	16	H'FFFFD202	16
		LAFM0	—	16	H'FFFFD204	16, 32
		LAFM1	—	16	H'FFFFD206	16
		DATA_01	—	16	H'FFFFD208	8, 16, 32
		DATA_23	—	16	H'FFFFD20A	8, 16
		DATA_45	—	16	H'FFFFD20C	8, 16, 32
		DATA_67	—	16	H'FFFFD20E	8, 16
		CONTROL1	—	16	H'FFFFD210	8, 16
		TIMESTAMP	—	16	H'FFFFD212	16
	MB[9].	CONTROL0_H	—	16	H'FFFFD220	16, 32
		CONTROL0_L	—	16	H'FFFFD222	16
		LAFM0	—	16	H'FFFFD224	16, 32
		LAFM1	—	16	H'FFFFD226	16

Module	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1 (RCAN_A)	MB[9].	DATA_01	—	16	H'FFFFD228	8, 16, 32
		DATA_23	—	16	H'FFFFD22A	8, 16
		DATA_45	—	16	H'FFFFD22C	8, 16, 32
		DATA_67	—	16	H'FFFFD22E	8, 16
		CONTROL1	—	16	H'FFFFD230	8, 16
		TIMESTAMP	—	16	H'FFFFD232	16
	MB[10].	CONTROL0_H	—	16	H'FFFFD240	16, 32
		CONTROL0_L	—	16	H'FFFFD242	16
		LAFM0	—	16	H'FFFFD244	16, 32
		LAFM1	—	16	H'FFFFD246	16
		DATA_01	—	16	H'FFFFD248	8, 16, 32
		DATA_23	—	16	H'FFFFD24A	8, 16
		DATA_45	—	16	H'FFFFD24C	8, 16, 32
		DATA_67	—	16	H'FFFFD24E	8, 16
		CONTROL1	—	16	H'FFFFD250	8, 16
		TIMESTAMP	—	16	H'FFFFD252	16
	MB[11].	CONTROL0_H	—	16	H'FFFFD260	16, 32
		CONTROL0_L	—	16	H'FFFFD262	16
		LAFM0	—	16	H'FFFFD264	16, 32
		LAFM1	—	16	H'FFFFD266	16
		DATA_01	—	16	H'FFFFD268	8, 16, 32
		DATA_23	—	16	H'FFFFD26A	8, 16
		DATA_45	—	16	H'FFFFD26C	8, 16, 32
		DATA_67	—	16	H'FFFFD26E	8, 16
		CONTROL1	—	16	H'FFFFD270	8, 16
		TIMESTAMP	—	16	H'FFFFD272	16
	MB[12].	CONTROL0_H	—	16	H'FFFFD280	16, 32
		CONTROL0_L	—	16	H'FFFFD282	16
		LAFM0	—	16	H'FFFFD284	16, 32
		LAFM1	—	16	H'FFFFD286	16

Module	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1 (RCAN_A)	MB[12].	DATA_01	—	16	H'FFFFD288	8, 16, 32
		DATA_23	—	16	H'FFFFD28A	8, 16
		DATA_45	—	16	H'FFFFD28C	8, 16, 32
		DATA_67	—	16	H'FFFFD28E	8, 16
		CONTROL1	—	16	H'FFFFD290	8, 16
		TIMESTAMP	—	16	H'FFFFD292	16
	MB[13].	CONTROL0_H	—	16	H'FFFFD2A0	16, 32
		CONTROL0_L	—	16	H'FFFFD2A2	16
		LAFM0	—	16	H'FFFFD2A4	16, 32
		LAFM1	—	16	H'FFFFD2A6	16
		DATA_01	—	16	H'FFFFD2A8	8, 16, 32
		DATA_23	—	16	H'FFFFD2AA	8, 16
		DATA_45	—	16	H'FFFFD2AC	8, 16, 32
		DATA_67	—	16	H'FFFFD2AE	8, 16
		CONTROL1	—	16	H'FFFFD2B0	8, 16
		TIMESTAMP	—	16	H'FFFFD2B2	16
	MB[14].	CONTROL0_H	—	16	H'FFFFD2C0	16, 32
		CONTROL0_L	—	16	H'FFFFD2C2	16
		LAFM0	—	16	H'FFFFD2C4	16, 32
		LAFM1	—	16	H'FFFFD2C6	16
		DATA_01	—	16	H'FFFFD2C8	8, 16, 32
		DATA_23	—	16	H'FFFFD2CA	8, 16
		DATA_45	—	16	H'FFFFD2CC	8, 16, 32
		DATA_67	—	16	H'FFFFD2CE	8, 16
		CONTROL1	—	16	H'FFFFD2D0	8, 16
		TIMESTAMP	—	16	H'FFFFD2D2	16
	MB[15].	CONTROL0_H	—	16	H'FFFFD2E0	16, 32
		CONTROL0_L	—	16	H'FFFFD2E2	16
		LAFM0	—	16	H'FFFFD2E4	16, 32
		LAFM1	—	16	H'FFFFD2E6	16

Module	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1 (RCAN_A)	MB[15].	DATA_01	—	16	H'FFFFD2E8	8, 16, 32
		DATA_23	—	16	H'FFFFD2EA	8, 16
		DATA_45	—	16	H'FFFFD2EC	8, 16, 32
		DATA_67	—	16	H'FFFFD2EE	8, 16
		CONTROL1	—	16	H'FFFFD2F0	8, 16
		TIMESTAMP	—	16	H'FFFFD2F2	16
	MB[16].	CONTROL0_H	—	16	H'FFFFD300	16, 32
		CONTROL0_L	—	16	H'FFFFD302	16
		LAFM0	—	16	H'FFFFD304	16, 32
		LAFM1	—	16	H'FFFFD306	16
		DATA_01	—	16	H'FFFFD308	8, 16, 32
		DATA_23	—	16	H'FFFFD30A	8, 16
		DATA_45	—	16	H'FFFFD30C	8, 16, 32
		DATA_67	—	16	H'FFFFD30E	8, 16
		CONTROL1	—	16	H'FFFFD310	8, 16
	MB[17].	CONTROL0_H	—	16	H'FFFFD320	16, 32
		CONTROL0_L	—	16	H'FFFFD322	16
		LAFM0	—	16	H'FFFFD324	16, 32
		LAFM1	—	16	H'FFFFD326	16
		DATA_01	—	16	H'FFFFD328	8, 16, 32
		DATA_23	—	16	H'FFFFD32A	8, 16
		DATA_45	—	16	H'FFFFD32C	8, 16, 32
		DATA_67	—	16	H'FFFFD32E	8, 16
		CONTROL1	—	16	H'FFFFD330	8, 16
	MB[18].	CONTROL0_H	—	16	H'FFFFD340	16, 32
		CONTROL0_L	—	16	H'FFFFD342	16
		LAFM0	—	16	H'FFFFD344	16, 32
		LAFM1	—	16	H'FFFFD346	16
		DATA_01	—	16	H'FFFFD348	8, 16, 32
		DATA_23	—	16	H'FFFFD34A	8, 16

Module	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1 (RCAN_A)	MB[18].	DATA_45	—	16	H'FFFFD34C	8, 16, 32
		DATA_67	—	16	H'FFFFD34E	8, 16
		CONTROL1	—	16	H'FFFFD350	8, 16
	MB[19].	CONTROL0_H	—	16	H'FFFFD360	16, 32
		CONTROL0_L	—	16	H'FFFFD362	16
		LAFM0	—	16	H'FFFFD364	16, 32
		LAFM1	—	16	H'FFFFD366	16
		DATA_01	—	16	H'FFFFD368	8, 16, 32
		DATA_23	—	16	H'FFFFD36A	8, 16
		DATA_45	—	16	H'FFFFD36C	8, 16, 32
		DATA_67	—	16	H'FFFFD36E	8, 16
		CONTROL1	—	16	H'FFFFD370	8, 16
	MB[20].	CONTROL0_H	—	16	H'FFFFD380	16, 32
		CONTROL0_L	—	16	H'FFFFD382	16
		LAFM0	—	16	H'FFFFD384	16, 32
		LAFM1	—	16	H'FFFFD386	16
		DATA_01	—	16	H'FFFFD388	8, 16, 32
		DATA_23	—	16	H'FFFFD38A	8, 16
		DATA_45	—	16	H'FFFFD38C	8, 16, 32
		DATA_67	—	16	H'FFFFD38E	8, 16
		CONTROL1	—	16	H'FFFFD390	8, 16
	MB[21].	CONTROL0_H	—	16	H'FFFFD3A0	16, 32
		CONTROL0_L	—	16	H'FFFFD3A2	16
		LAFM0	—	16	H'FFFFD3A4	16, 32
		LAFM1	—	16	H'FFFFD3A6	16
		DATA_01	—	16	H'FFFFD3A8	8, 16, 32
		DATA_23	—	16	H'FFFFD3AA	8, 16
		DATA_45	—	16	H'FFFFD3AC	8, 16, 32
		DATA_67	—	16	H'FFFFD3AE	8, 16
		CONTROL1	—	16	H'FFFFD3B0	8, 16

Module	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1 (RCAN_A)	MB[22].	CONTROL0_H	—	16	H'FFFFD3C0	16, 32
		CONTROL0_L	—	16	H'FFFFD3C2	16
		LAFM0	—	16	H'FFFFD3C4	16, 32
		LAFM1	—	16	H'FFFFD3C6	16
		DATA_01	—	16	H'FFFFD3C8	8, 16, 32
		DATA_23	—	16	H'FFFFD3CA	8, 16
		DATA_45	—	16	H'FFFFD3CC	8, 16, 32
		DATA_67	—	16	H'FFFFD3CE	8, 16
		CONTROL1	—	16	H'FFFFD3D0	8, 16
	MB[23].	CONTROL0_H	—	16	H'FFFFD3E0	16, 32
		CONTROL0_L	—	16	H'FFFFD3E2	16
		LAFM0	—	16	H'FFFFD3E4	16, 32
		LAFM1	—	16	H'FFFFD3E6	16
		DATA_01	—	16	H'FFFFD3E8	8, 16, 32
		DATA_23	—	16	H'FFFFD3EA	8, 16
		DATA_45	—	16	H'FFFFD3EC	8, 16, 32
		DATA_67	—	16	H'FFFFD3EE	8, 16
		CONTROL1	—	16	H'FFFFD3F0	8, 16
	MB[24].	CONTROL0_H	—	16	H'FFFFD400	16, 32
		CONTROL0_L	—	16	H'FFFFD402	16
		LAFM0	—	16	H'FFFFD404	16, 32
		LAFM1	—	16	H'FFFFD406	16
		DATA_01	—	16	H'FFFFD408	8, 16, 32
		DATA_23	—	16	H'FFFFD40A	8, 16
		DATA_45	—	16	H'FFFFD40C	8, 16, 32
		DATA_67	—	16	H'FFFFD40E	8, 16
		CONTROL1	—	16	H'FFFFD410	8, 16
		TTT	—	16	H'FFFFD414	16
		TTCONTROL	—	16	H'FFFFD416	16
	MB[25].	CONTROL0_H	—	16	H'FFFFD420	16, 32

Module	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1 (RCAN_A)	MB[25].	CONTROL0_L	—	16	H'FFFFD422	16
		LAFM0	—	16	H'FFFFD424	16, 32
		LAFM1	—	16	H'FFFFD426	16
		DATA_01	—	16	H'FFFFD428	8, 16, 32
		DATA_23	—	16	H'FFFFD42A	8, 16
		DATA_45	—	16	H'FFFFD42C	8, 16, 32
		DATA_67	—	16	H'FFFFD42E	8, 16
		CONTROL1	—	16	H'FFFFD430	8, 16
		TTT	—	16	H'FFFFD434	16
		TTCONTROL	—	16	H'FFFFD436	16
	MB[26].	CONTROL0_H	—	16	H'FFFFD440	16, 32
		CONTROL0_L	—	16	H'FFFFD442	16
		LAFM0	—	16	H'FFFFD444	16, 32
		LAFM1	—	16	H'FFFFD446	16
		DATA_01	—	16	H'FFFFD448	8, 16, 32
		DATA_23	—	16	H'FFFFD44A	8, 16
		DATA_45	—	16	H'FFFFD44C	8, 16, 32
		DATA_67	—	16	H'FFFFD44E	8, 16
		CONTROL1	—	16	H'FFFFD450	8, 16
		TTT	—	16	H'FFFFD454	16
		TTCONTROL	—	16	H'FFFFD456	16
	MB[27].	CONTROL0_H	—	16	H'FFFFD460	16, 32
		CONTROL0_L	—	16	H'FFFFD462	16
		LAFM0	—	16	H'FFFFD464	16, 32
		LAFM1	—	16	H'FFFFD466	16
		DATA_01	—	16	H'FFFFD468	8, 16, 32
		DATA_23	—	16	H'FFFFD46A	8, 16
		DATA_45	—	16	H'FFFFD46C	8, 16, 32
		DATA_67	—	16	H'FFFFD46E	8, 16
		CONTROL1	—	16	H'FFFFD470	8, 16

Module	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1 (RCAN_A)	MB[27].	TTT	—	16	H'FFFFD474	16
		TTCONTROL	—	16	H'FFFFD476	16
	MB[28].	CONTROL0_H	—	16	H'FFFFD480	16, 32
		CONTROL0_L	—	16	H'FFFFD482	16
		LAFM0	—	16	H'FFFFD484	16, 32
		LAFM1	—	16	H'FFFFD486	16
		DATA_01	—	16	H'FFFFD488	8, 16, 32
		DATA_23	—	16	H'FFFFD48A	8, 16
		DATA_45	—	16	H'FFFFD48C	8, 16, 32
		DATA_67	—	16	H'FFFFD48E	8, 16
		CONTROL1	—	16	H'FFFFD490	8, 16
		TTT	—	16	H'FFFFD494	16
		TTCONTROL	—	16	H'FFFFD496	16
	MB[29].	CONTROL0_H	—	16	H'FFFFD4A0	16, 32
		CONTROL0_L	—	16	H'FFFFD4A2	16
		LAFM0	—	16	H'FFFFD4A4	16, 32
		LAFM1	—	16	H'FFFFD4A6	16
		DATA_01	—	16	H'FFFFD4A8	8, 16, 32
		DATA_23	—	16	H'FFFFD4AA	8, 16
		DATA_45	—	16	H'FFFFD4AC	8, 16, 32
		DATA_67	—	16	H'FFFFD4AE	8, 16
		CONTROL1	—	16	H'FFFFD4B0	8, 16
		TTT	—	16	H'FFFFD4B4	16
		TTCONTROL	—	16	H'FFFFD4B6	16
	MB[30].	CONTROL0_H	—	16	H'FFFFD4C0	16, 32
		CONTROL0_L	—	16	H'FFFFD4C2	16
		LAFM0	—	16	H'FFFFD4C4	16, 32
		LAFM1	—	16	H'FFFFD4C6	16
		DATA_01	—	16	H'FFFFD4C8	8, 16, 32
		DATA_23	—	16	H'FFFFD4CA	8, 16

Module	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1 (RCAN_A)	MB[30].	DATA_45	—	16	H'FFFFD4CC	8, 16, 32
		DATA_67	—	16	H'FFFFD4CE	8, 16
		CONTROL1	—	16	H'FFFFD4D0	8, 16
		TIMESTAMP	—	16	H'FFFFD4D2	16
		TTT	—	16	H'FFFFD4D4	16
	MB[31].	CONTROL0_H	—	16	H'FFFFD4E0	16, 32
		CONTROL0_L	—	16	H'FFFFD4E2	16
		LAFM0	—	16	H'FFFFD4E4	16, 32
		LAFM1	—	16	H'FFFFD4E6	16
		DATA_01	—	16	H'FFFFD4E8	8, 16, 32
		DATA_23	—	16	H'FFFFD4EA	8, 16
		DATA_45	—	16	H'FFFFD4EC	8, 16, 32
		DATA_67	—	16	H'FFFFD4EE	8, 16
		CONTROL1	—	16	H'FFFFD4F0	8, 16
		TIMESTAMP	—	16	H'FFFFD4F2	16
	Message buffer error status register		MBESR	16	H'FFFFD600	16
	Message buffer error control register		MBECR	16	H'FFFFD602	16
RCAN-TL1 (RCAN_B)	Master control register		MCR	16	H'FFFFD800	16
	General status register		GSR	16	H'FFFFD802	16
	Bit configuration register 1		BCR1	16	H'FFFFD804	16
	Bit configuration register 0		BCR0	16	H'FFFFD806	16
	Interrupt request register		IRR	16	H'FFFFD808	16
	Interrupt mask register		IMR	16	H'FFFFD80A	16
	Error counter register		TEC/REC	16	H'FFFFD80C	16
	Transmit pending register 1		TXPR1	16	H'FFFFD820	32
	Transmit pending register 0		TXPR0	16		
	Transmit cancel register 1		TXCR1	16	H'FFFFD828	16
	Transmit cancel register 0		TXCR0	16	H'FFFFD82A	16

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1 (RCAN_B)	Transmit acknowledge register 1	TXACK1	16	H'FFFFD830	16
	Transmit acknowledge register 0	TXACK0	16	H'FFFFD832	16
	Abort acknowledge register 1	ABACK1	16	H'FFFFD838	16
	Abort acknowledge register 0	ABACK0	16	H'FFFFD83A	16
	Data frame receive pending register 1	RXPR1	16	H'FFFFD840	16
	Data frame receive pending register 0	RXPR0	16	H'FFFFD842	16
	Remote frame receive pending register 1	RFPR1	16	H'FFFFD848	16
	Remote frame receive pending register 0	RFPR0	16	H'FFFFD84A	16
	Mailbox interrupt mask register 1	MBIMR1	16	H'FFFFD850	16
	Mailbox interrupt mask register 0	MBIMR0	16	H'FFFFD852	16
	Unread message status register 1	UMSR1	16	H'FFFFD858	16
	Unread message status register 0	UMSR0	16	H'FFFFD85A	16
	Timer trigger control register 0	TTCR0	16	H'FFFFD880	16
	Cycle maximum/tx-enable window register	CMAX_TEW	16	H'FFFFD884	16
	Reference trigger offset register	RFTROFF	16	H'FFFFD886	16
	Timer status register	TSR	16	H'FFFFD888	16
	Cycle counter register	CCR	16	H'FFFFD88A	16
	Timer counter register	TCNTR	16	H'FFFFD88C	16
	Cycle time register	CYCTR	16	H'FFFFD890	16
	Reference mark register	RFMK	16	H'FFFFD894	16
	Timer compare match register 0	TCMR0	16	H'FFFFD898	16
	Timer compare match register 1	TCMR1	16	H'FFFFD89C	16
	Timer compare match register 2	TCMR2	16	H'FFFFD8A0	16
	Tx-trigger time selection register	TTTSEL	16	H'FFFFD8A4	16

Module	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1 (RCAN_B)	MB[0].	CONTROL0_H	—	16	H'FFFFD900	16, 32
		CONTROL0_L	—	16	H'FFFFD902	16
		LAFM0	—	16	H'FFFFD904	16, 32
		LAFM1	—	16	H'FFFFD906	16
		DATA_01	—	16	H'FFFFD908	8, 16, 32
		DATA_23	—	16	H'FFFFD90A	8, 16
		DATA_45	—	16	H'FFFFD90C	8, 16, 32
		DATA_67	—	16	H'FFFFD90E	8, 16
		CONTROL1	—	16	H'FFFFD910	8, 16
		TIMESTAMP	—	16	H'FFFFD912	16
	MB[1].	CONTROL0_H	—	16	H'FFFFD920	16, 32
		CONTROL0_L	—	16	H'FFFFD922	16
		LAFM0	—	16	H'FFFFD924	16, 32
		LAFM1	—	16	H'FFFFD926	16
		DATA_01	—	16	H'FFFFD928	8, 16, 32
		DATA_23	—	16	H'FFFFD92A	8, 16
		DATA_45	—	16	H'FFFFD92C	8, 16, 32
		DATA_67	—	16	H'FFFFD92E	8, 16
		CONTROL1	—	16	H'FFFFD930	8, 16
		TIMESTAMP	—	16	H'FFFFD932	16
	MB[2].	CONTROL0_H	—	16	H'FFFFD940	16, 32
		CONTROL0_L	—	16	H'FFFFD942	16
		LAFM0	—	16	H'FFFFD944	16, 32
		LAFM1	—	16	H'FFFFD946	16
		DATA_01	—	16	H'FFFFD948	8, 16, 32
		DATA_23	—	16	H'FFFFD94A	8, 16
		DATA_45	—	16	H'FFFFD94C	8, 16, 32
		DATA_67	—	16	H'FFFFD94E	8, 16
		CONTROL1	—	16	H'FFFFD950	8, 16
		TIMESTAMP	—	16	H'FFFFD952	16

Module	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1 (RCAN_B)	MB[3].	CONTROL0_H	—	16	H'FFFFD960	16, 32
		CONTROL0_L	—	16	H'FFFFD962	16
		LAFM0	—	16	H'FFFFD964	16, 32
		LAFM1	—	16	H'FFFFD966	16
		DATA_01	—	16	H'FFFFD968	8, 16, 32
		DATA_23	—	16	H'FFFFD96A	8, 16
		DATA_45	—	16	H'FFFFD96C	8, 16, 32
		DATA_67	—	16	H'FFFFD96E	8, 16
		CONTROL1	—	16	H'FFFFD970	8, 16
		TIMESTAMP	—	16	H'FFFFD972	16
	MB[4].	CONTROL0_H	—	16	H'FFFFD980	16, 32
		CONTROL0_L	—	16	H'FFFFD982	16
		LAFM0	—	16	H'FFFFD984	16, 32
		LAFM1	—	16	H'FFFFD986	16
		DATA_01	—	16	H'FFFFD988	8, 16, 32
		DATA_23	—	16	H'FFFFD98A	8, 16
		DATA_45	—	16	H'FFFFD98C	8, 16, 32
		DATA_67	—	16	H'FFFFD98E	8, 16
		CONTROL1	—	16	H'FFFFD990	8, 16
		TIMESTAMP	—	16	H'FFFFD992	16
	MB[5].	CONTROL0_H	—	16	H'FFFFD9A0	16, 32
		CONTROL0_L	—	16	H'FFFFD9A2	16
		LAFM0	—	16	H'FFFFD9A4	16, 32
		LAFM1	—	16	H'FFFFD9A6	16
		DATA_01	—	16	H'FFFFD9A8	8, 16, 32
		DATA_23	—	16	H'FFFFD9AA	8, 16
		DATA_45	—	16	H'FFFFD9AC	8, 16, 32
		DATA_67	—	16	H'FFFFD9AE	8, 16
		CONTROL1	—	16	H'FFFFD9B0	8, 16
		TIMESTAMP	—	16	H'FFFFD9B2	16

Module	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1 (RCAN_B)	MB[6].	CONTROL0_H	—	16	H'FFFFD9C0	16, 32
		CONTROL0_L	—	16	H'FFFFD9C2	16
		LAFM0	—	16	H'FFFFD9C4	16, 32
		LAFM1	—	16	H'FFFFD9C6	16
		DATA_01	—	16	H'FFFFD9C8	8, 16, 32
		DATA_23	—	16	H'FFFFD9CA	8, 16
		DATA_45	—	16	H'FFFFD9CC	8, 16, 32
		DATA_67	—	16	H'FFFFD9CE	8, 16
		CONTROL1	—	16	H'FFFFD9D0	8, 16
		TIMESTAMP	—	16	H'FFFFD9D2	16
	MB[7].	CONTROL0_H	—	16	H'FFFFD9E0	16, 32
		CONTROL0_L	—	16	H'FFFFD9E2	16
		LAFM0	—	16	H'FFFFD9E4	16, 32
		LAFM1	—	16	H'FFFFD9E6	16
		DATA_01	—	16	H'FFFFD9E8	8, 16, 32
		DATA_23	—	16	H'FFFFD9EA	8, 16
		DATA_45	—	16	H'FFFFD9EC	8, 16, 32
		DATA_67	—	16	H'FFFFD9EE	8, 16
		CONTROL1	—	16	H'FFFFD9F0	8, 16
		TIMESTAMP	—	16	H'FFFFD9F2	16
	MB[8].	CONTROL0_H	—	16	H'FFFFDA00	16, 32
		CONTROL0_L	—	16	H'FFFFDA02	16
		LAFM0	—	16	H'FFFFDA04	16, 32
		LAFM1	—	16	H'FFFFDA06	16
		DATA_01	—	16	H'FFFFDA08	8, 16, 32
		DATA_23	—	16	H'FFFFDA0A	8, 16
		DATA_45	—	16	H'FFFFDA0C	8, 16, 32
		DATA_67	—	16	H'FFFFDA0E	8, 16
		CONTROL1	—	16	H'FFFFDA10	8, 16
		TIMESTAMP	—	16	H'FFFFDA12	16

Module	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1 (RCAN_B)	MB[9].	CONTROL0_H	—	16	H'FFFFDA20	16, 32
		CONTROL0_L	—	16	H'FFFFDA22	16
		LAFM0	—	16	H'FFFFDA24	16, 32
		LAFM1	—	16	H'FFFFDA26	16
		DATA_01	—	16	H'FFFFDA28	8, 16, 32
		DATA_23	—	16	H'FFFFDA2A	8, 16
		DATA_45	—	16	H'FFFFDA2C	8, 16, 32
		DATA_67	—	16	H'FFFFDA2E	8, 16
		CONTROL1	—	16	H'FFFFDA30	8, 16
		TIMESTAMP	—	16	H'FFFFDA32	16
	MB[10].	CONTROL0_H	—	16	H'FFFFDA40	16, 32
		CONTROL0_L	—	16	H'FFFFDA42	16
		LAFM0	—	16	H'FFFFDA44	16, 32
		LAFM1	—	16	H'FFFFDA46	16
		DATA_01	—	16	H'FFFFDA48	8, 16, 32
		DATA_23	—	16	H'FFFFDA4A	8, 16
		DATA_45	—	16	H'FFFFDA4C	8, 16, 32
		DATA_67	—	16	H'FFFFDA4E	8, 16
		CONTROL1	—	16	H'FFFFDA50	8, 16
		TIMESTAMP	—	16	H'FFFFDA52	16
	MB[11].	CONTROL0_H	—	16	H'FFFFDA60	16, 32
		CONTROL0_L	—	16	H'FFFFDA62	16
		LAFM0	—	16	H'FFFFDA64	16, 32
		LAFM1	—	16	H'FFFFDA66	16
		DATA_01	—	16	H'FFFFDA68	8, 16, 32
		DATA_23	—	16	H'FFFFDA6A	8, 16
		DATA_45	—	16	H'FFFFDA6C	8, 16, 32
		DATA_67	—	16	H'FFFFDA6E	8, 16
		CONTROL1	—	16	H'FFFFDA70	8, 16
		TIMESTAMP	—	16	H'FFFFDA72	16

Module	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1 (RCAN_B)	MB[12].	CONTROL0_H	—	16	H'FFFFDA80	16, 32
		CONTROL0_L	—	16	H'FFFFDA82	16
		LAFM0	—	16	H'FFFFDA84	16, 32
		LAFM1	—	16	H'FFFFDA86	16
		DATA_01	—	16	H'FFFFDA88	8, 16, 32
		DATA_23	—	16	H'FFFFDA8A	8, 16
		DATA_45	—	16	H'FFFFDA8C	8, 16, 32
		DATA_67	—	16	H'FFFFDA8E	8, 16
		CONTROL1	—	16	H'FFFFDA90	8, 16
		TIMESTAMP	—	16	H'FFFFDA92	16
	MB[13].	CONTROL0_H	—	16	H'FFFFDAA0	16, 32
		CONTROL0_L	—	16	H'FFFFDAA2	16
		LAFM0	—	16	H'FFFFDAA4	16, 32
		LAFM1	—	16	H'FFFFDAA6	16
		DATA_01	—	16	H'FFFFDAA8	8, 16, 32
		DATA_23	—	16	H'FFFFDAAA	8, 16
		DATA_45	—	16	H'FFFFDAAC	8, 16, 32
		DATA_67	—	16	H'FFFFDAAE	8, 16
		CONTROL1	—	16	H'FFFFDAB0	8, 16
		TIMESTAMP	—	16	H'FFFFDAB2	16
	MB[14].	CONTROL0_H	—	16	H'FFFFDAC0	16, 32
		CONTROL0_L	—	16	H'FFFFDAC2	16
		LAFM0	—	16	H'FFFFDAC4	16, 32
		LAFM1	—	16	H'FFFFDAC6	16
		DATA_01	—	16	H'FFFFDAC8	8, 16, 32
		DATA_23	—	16	H'FFFFDACA	8, 16
		DATA_45	—	16	H'FFFFDACC	8, 16, 32
		DATA_67	—	16	H'FFFFDACE	8, 16
		CONTROL1	—	16	H'FFFFDAD0	8, 16
		TIMESTAMP	—	16	H'FFFFDAD2	16

Module	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1 (RCAN_B)	MB[15].	CONTROL0_H	—	16	H'FFFFDAE0	16, 32
		CONTROL0_L	—	16	H'FFFFDAE2	16
		LAFM0	—	16	H'FFFFDAE4	16, 32
		LAFM1	—	16	H'FFFFDAE6	16
		DATA_01	—	16	H'FFFFDAE8	8, 16, 32
		DATA_23	—	16	H'FFFFDAEA	8, 16
		DATA_45	—	16	H'FFFFDAEC	8, 16, 32
		DATA_67	—	16	H'FFFFDAEE	8, 16
		CONTROL1	—	16	H'FFFFDAF0	8, 16
		TIMESTAMP	—	16	H'FFFFDAF2	16
	MB[16].	CONTROL0_H	—	16	H'FFFFDB00	16, 32
		CONTROL0_L	—	16	H'FFFFDB02	16
		LAFM0	—	16	H'FFFFDB04	16, 32
		LAFM1	—	16	H'FFFFDB06	16
		DATA_01	—	16	H'FFFFDB08	8, 16, 32
		DATA_23	—	16	H'FFFFDB0A	8, 16
		DATA_45	—	16	H'FFFFDB0C	8, 16, 32
		DATA_67	—	16	H'FFFFDB0E	8, 16
		CONTROL1	—	16	H'FFFFDB10	8, 16
	MB[17].	CONTROL0_H	—	16	H'FFFFDB20	16, 32
		CONTROL0_L	—	16	H'FFFFDB22	16
		LAFM0	—	16	H'FFFFDB24	16, 32
		LAFM1	—	16	H'FFFFDB26	16
		DATA_01	—	16	H'FFFFDB28	8, 16, 32
		DATA_23	—	16	H'FFFFDB2A	8, 16
		DATA_45	—	16	H'FFFFDB2C	8, 16, 32
		DATA_67	—	16	H'FFFFDB2E	8, 16
		CONTROL1	—	16	H'FFFFDB30	8, 16
	MB[18].	CONTROL0_H	—	16	H'FFFFDB40	16, 32
		CONTROL0_L	—	16	H'FFFFDB42	16
		LAFM0	—	16	H'FFFFDB44	16, 32

Module	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1 (RCAN_B)	MB[18].	LAFM1	—	16	H'FFFFDB46	16
		DATA_01	—	16	H'FFFFDB48	8, 16, 32
		DATA_23	—	16	H'FFFFDB4A	8, 16
		DATA_45	—	16	H'FFFFDB4C	8, 16, 32
		DATA_67	—	16	H'FFFFDB4E	8, 16
		CONTROL1	—	16	H'FFFFDB50	8, 16
	MB[19].	CONTROL0_H	—	16	H'FFFFDB60	16, 32
		CONTROL0_L	—	16	H'FFFFDB62	16
		LAFM0	—	16	H'FFFFDB64	16, 32
		LAFM1	—	16	H'FFFFDB66	16
		DATA_01	—	16	H'FFFFDB68	8, 16, 32
		DATA_23	—	16	H'FFFFDB6A	8, 16
		DATA_45	—	16	H'FFFFDB6C	8, 16, 32
		DATA_67	—	16	H'FFFFDB6E	8, 16
		CONTROL1	—	16	H'FFFFDB70	8, 16
	MB[20].	CONTROL0_H	—	16	H'FFFFDB80	16, 32
		CONTROL0_L	—	16	H'FFFFDB82	16
		LAFM0	—	16	H'FFFFDB84	16, 32
		LAFM1	—	16	H'FFFFDB86	16
		DATA_01	—	16	H'FFFFDB88	8, 16, 32
		DATA_23	—	16	H'FFFFDB8A	8, 16
		DATA_45	—	16	H'FFFFDB8C	8, 16, 32
		DATA_67	—	16	H'FFFFDB8E	8, 16
		CONTROL1	—	16	H'FFFFDB90	8, 16
	MB[21].	CONTROL0_H	—	16	H'FFFFDBA0	16, 32
		CONTROL0_L	—	16	H'FFFFDBA2	16
		LAFM0	—	16	H'FFFFDBA4	16, 32
		LAFM1	—	16	H'FFFFDBA6	16
		DATA_01	—	16	H'FFFFDBA8	8, 16, 32
		DATA_23	—	16	H'FFFFDBAA	8, 16
		DATA_45	—	16	H'FFFFDBAC	8, 16, 32

Module	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1 (RCAN_B)	MB[21].	DATA_67	—	16	H'FFFFDBAE	8, 16
		CONTROL1	—	16	H'FFFFDBB0	8, 16
	MB[22].	CONTROL0_H	—	16	H'FFFFDBC0	16, 32
		CONTROL0_L	—	16	H'FFFFDBC2	16
		LAFM0	—	16	H'FFFFDBC4	16, 32
		LAFM1	—	16	H'FFFFDBC6	16
		DATA_01	—	16	H'FFFFDBC8	8, 16, 32
		DATA_23	—	16	H'FFFFDBCA	8, 16
		DATA_45	—	16	H'FFFFDBCC	8, 16, 32
		DATA_67	—	16	H'FFFFDBCE	8, 16
		CONTROL1	—	16	H'FFFFDBD0	8, 16
	MB[23].	CONTROL0_H	—	16	H'FFFFDBE0	16, 32
		CONTROL0_L	—	16	H'FFFFDBE2	16
		LAFM0	—	16	H'FFFFDBE4	16, 32
		LAFM1	—	16	H'FFFFDBE6	16
		DATA_01	—	16	H'FFFFDBE8	8, 16, 32
		DATA_23	—	16	H'FFFFDBEA	8, 16
		DATA_45	—	16	H'FFFFDBEC	8, 16, 32
		DATA_67	—	16	H'FFFFDBEE	8, 16
		CONTROL1	—	16	H'FFFFDBF0	8, 16
	MB[24].	CONTROL0_H	—	16	H'FFFFDC00	16, 32
		CONTROL0_L	—	16	H'FFFFDC02	16
		LAFM0	—	16	H'FFFFDC04	16, 32
		LAFM1	—	16	H'FFFFDC06	16
		DATA_01	—	16	H'FFFFDC08	8, 16, 32
		DATA_23	—	16	H'FFFFDC0A	8, 16
		DATA_45	—	16	H'FFFFDC0C	8, 16, 32
		DATA_67	—	16	H'FFFFDC0E	8, 16
		CONTROL1	—	16	H'FFFFDC10	8, 16
		TTT	—	16	H'FFFFDC14	16
		TTCONTROL	—	16	H'FFFFDC16	16

Module	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1 (RCAN_B)	MB[25].	CONTROL0_H	—	16	H'FFFFDC20	16, 32
		CONTROL0_L	—	16	H'FFFFDC22	16
		LAFM0	—	16	H'FFFFDC24	16, 32
		LAFM1	—	16	H'FFFFDC26	16
		DATA_01	—	16	H'FFFFDC28	8, 16, 32
		DATA_23	—	16	H'FFFFDC2A	8, 16
		DATA_45	—	16	H'FFFFDC2C	8, 16, 32
		DATA_67	—	16	H'FFFFDC2E	8, 16
		CONTROL1	—	16	H'FFFFDC30	8, 16
		TTT	—	16	H'FFFFDC34	16
		TTCONTROL	—	16	H'FFFFDC36	16
	MB[26].	CONTROL0_H	—	16	H'FFFFDC40	16, 32
		CONTROL0_L	—	16	H'FFFFDC42	16
		LAFM0	—	16	H'FFFFDC44	16, 32
		LAFM1	—	16	H'FFFFDC46	16
		DATA_01	—	16	H'FFFFDC48	8, 16, 32
		DATA_23	—	16	H'FFFFDC4A	8, 16
		DATA_45	—	16	H'FFFFDC4C	8, 16, 32
		DATA_67	—	16	H'FFFFDC4E	8, 16
		CONTROL1	—	16	H'FFFFDC50	8, 16
		TTT	—	16	H'FFFFDC54	16
		TTCONTROL	—	16	H'FFFFDC56	16
	MB[27].	CONTROL0_H	—	16	H'FFFFDC60	16, 32
		CONTROL0_L	—	16	H'FFFFDC62	16
		LAFM0	—	16	H'FFFFDC64	16, 32
		LAFM1	—	16	H'FFFFDC66	16
		DATA_01	—	16	H'FFFFDC68	8, 16, 32
		DATA_23	—	16	H'FFFFDC6A	8, 16
		DATA_45	—	16	H'FFFFDC6C	8, 16, 32
		DATA_67	—	16	H'FFFFDC6E	8, 16

Module	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1 (RCAN_B)	MB[27].	CONTROL1	—	16	H'FFFFDC70	8, 16
		TTT	—	16	H'FFFFDC74	16
		TTCONTROL	—	16	H'FFFFDC76	16
	MB[28].	CONTROL0_H	—	16	H'FFFFDC80	16, 32
		CONTROL0_L	—	16	H'FFFFDC82	16
		LAFM0	—	16	H'FFFFDC84	16, 32
		LAFM1	—	16	H'FFFFDC86	16
		DATA_01	—	16	H'FFFFDC88	8, 16, 32
		DATA_23	—	16	H'FFFFDC8A	8, 16
		DATA_45	—	16	H'FFFFDC8C	8, 16, 32
		DATA_67	—	16	H'FFFFDC8E	8, 16
		CONTROL1	—	16	H'FFFFDC90	8, 16
		TTT	—	16	H'FFFFDC94	16
		TTCONTROL	—	16	H'FFFFDC96	16
	MB[29].	CONTROL0_H	—	16	H'FFFFDCA0	16, 32
		CONTROL0_L	—	16	H'FFFFDCA2	16
		LAFM0	—	16	H'FFFFDCA4	16, 32
		LAFM1	—	16	H'FFFFDCA6	16
		DATA_01	—	16	H'FFFFDCA8	8, 16, 32
		DATA_23	—	16	H'FFFFDCAA	8, 16
		DATA_45	—	16	H'FFFFDCAC	8, 16, 32
		DATA_67	—	16	H'FFFFDCAE	8, 16
		CONTROL1	—	16	H'FFFFDCB0	8, 16
		TTT	—	16	H'FFFFDCB4	16
		TTCONTROL	—	16	H'FFFFDCB6	16
	MB[30].	CONTROL0_H	—	16	H'FFFFDCC0	16, 32
		CONTROL0_L	—	16	H'FFFFDCC2	16
		LAFM0	—	16	H'FFFFDCC4	16, 32
		LAFM1	—	16	H'FFFFDCC6	16
		DATA_01	—	16	H'FFFFDCC8	8, 16, 32

Module	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1 (RCAN_B)	MB[30].	DATA_23	—	16	H'FFFFDCCA	8, 16
		DATA_45	—	16	H'FFFFDCCC	8, 16, 32
		DATA_67	—	16	H'FFFFDCCE	8, 16
		CONTROL1	—	16	H'FFFFDCD0	8, 16
		TIMESTAMP	—	16	H'FFFFDCD2	16
		TTT	—	16	H'FFFFDCD4	16
	MB[31].	CONTROL0_H	—	16	H'FFFFDCE0	16, 32
		CONTROL0_L	—	16	H'FFFFDCE2	16
		LAFM0	—	16	H'FFFFDCE4	16, 32
		LAFM1	—	16	H'FFFFDCE6	16
		DATA_01	—	16	H'FFFFDCE8	8, 16, 32
		DATA_23	—	16	H'FFFFDCEA	8, 16
		DATA_45	—	16	H'FFFFDCEC	8, 16, 32
		DATA_67	—	16	H'FFFFDCEE	8, 16
		CONTROL1	—	16	H'FFFFDCF0	8, 16
		TIMESTAMP	—	16	H'FFFFDCF2	16
	Message buffer error status register		MBESR	16	H'FFFFDE00	16
	Message buffer error control register		MBECR	16	H'FFFFDE02	16
RCAN-TL1 (RCAN_C)	Master control register		MCR	16	H'FFFFE000	16
	General status register		GSR	16	H'FFFFE002	16
	Bit configuration register 1		BCR1	16	H'FFFFE004	16
	Bit configuration register 0		BCR0	16	H'FFFFE006	16
	Interrupt request register		IRR	16	H'FFFFE008	16
	Interrupt mask register		IMR	16	H'FFFFE00A	16
	Error counter register		TEC/REC	16	H'FFFFE00C	16
	Transmit pending register 1		TXPR1	16	H'FFFFE020	32
	Transmit pending register 0		TXPR0	16		
	Transmit cancel register 1		TXCR1	16	H'FFFFE028	16

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1 (RCAN_C)	Transmit cancel register 0	TXCR0	16	H'FFFFE02A	16
	Transmit acknowledge register 1	TXACK1	16	H'FFFFE030	16
	Transmit acknowledge register 0	TXACK0	16	H'FFFFE032	16
	Abort acknowledge register 1	ABACK1	16	H'FFFFE038	16
	Abort acknowledge register 0	ABACK0	16	H'FFFFE03A	16
	Data frame receive pending register 1	RXPR1	16	H'FFFFE040	16
	Data frame receive pending register 0	RXPR0	16	H'FFFFE042	16
	Remote frame receive pending register 1	RFPR1	16	H'FFFFE048	16
	Remote frame receive pending register 0	RFPR0	16	H'FFFFE04A	16
	Mailbox interrupt mask register 1	MBIMR1	16	H'FFFFE050	16
	Mailbox interrupt mask register 0	MBIMR0	16	H'FFFFE052	16
	Unread message status register 1	UMSR1	16	H'FFFFE058	16
	Unread message status register 0	UMSR0	16	H'FFFFE05A	16
	Timer trigger control register 0	TTCR0	16	H'FFFFE080	16
	Cycle maximum/tx-enable window register	CMAX_TEW	16	H'FFFFE084	16
	Reference trigger offset register	RFTROFF	16	H'FFFFE086	16
	Timer status register	TSR	16	H'FFFFE088	16
	Cycle counter register	CCR	16	H'FFFFE08A	16
	Timer counter register	TCNTR	16	H'FFFFE08C	16
	Cycle time register	CYCTR	16	H'FFFFE090	16
	Reference mark register	RFMK	16	H'FFFFE094	16
	Timer compare match register 0	TCMR0	16	H'FFFFE098	16
	Timer compare match register 1	TCMR1	16	H'FFFFE09C	16
	Timer compare match register 2	TCMR2	16	H'FFFFE0A0	16
	Tx-trigger time selection register	TTTSEL	16	H'FFFFE0A4	16

Module	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1 (RCAN_C)	MB[0].	CONTROL0_H	—	16	H'FFFFE100	16, 32
		CONTROL0_L	—	16	H'FFFFE102	16
		LAFM0	—	16	H'FFFFE104	16, 32
		LAFM1	—	16	H'FFFFE106	16
		DATA_01	—	16	H'FFFFE108	8, 16, 32
		DATA_23	—	16	H'FFFFE10A	8, 16
		DATA_45	—	16	H'FFFFE10C	8, 16, 32
		DATA_67	—	16	H'FFFFE10E	8, 16
		CONTROL1	—	16	H'FFFFE110	8, 16
		TIMESTAMP	—	16	H'FFFFE112	16
	MB[1].	CONTROL0_H	—	16	H'FFFFE120	16, 32
		CONTROL0_L	—	16	H'FFFFE122	16
		LAFM0	—	16	H'FFFFE124	16, 32
		LAFM1	—	16	H'FFFFE126	16
		DATA_01	—	16	H'FFFFE128	8, 16, 32
		DATA_23	—	16	H'FFFFE12A	8, 16
		DATA_45	—	16	H'FFFFE12C	8, 16, 32
		DATA_67	—	16	H'FFFFE12E	8, 16
		CONTROL1	—	16	H'FFFFE130	8, 16
		TIMESTAMP	—	16	H'FFFFE132	16
	MB[2].	CONTROL0_H	—	16	H'FFFFE140	16, 32
		CONTROL0_L	—	16	H'FFFFE142	16
		LAFM0	—	16	H'FFFFE144	16, 32
		LAFM1	—	16	H'FFFFE146	16
		DATA_01	—	16	H'FFFFE148	8, 16, 32
		DATA_23	—	16	H'FFFFE14A	8, 16
		DATA_45	—	16	H'FFFFE14C	8, 16, 32
		DATA_67	—	16	H'FFFFE14E	8, 16
		CONTROL1	—	16	H'FFFFE150	8, 16
		TIMESTAMP	—	16	H'FFFFE152	16

Module	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1 (RCAN_C)	MB[3].	CONTROL0_H	—	16	H'FFFFE160	16, 32
		CONTROL0_L	—	16	H'FFFFE162	16
		LAFM0	—	16	H'FFFFE164	16, 32
		LAFM1	—	16	H'FFFFE166	16
		DATA_01	—	16	H'FFFFE168	8, 16, 32
		DATA_23	—	16	H'FFFFE16A	8, 16
		DATA_45	—	16	H'FFFFE16C	8, 16, 32
		DATA_67	—	16	H'FFFFE16E	8, 16
		CONTROL1	—	16	H'FFFFE170	8, 16
		TIMESTAMP	—	16	H'FFFFE172	16
	MB[4].	CONTROL0_H	—	16	H'FFFFE180	16, 32
		CONTROL0_L	—	16	H'FFFFE182	16
		LAFM0	—	16	H'FFFFE184	16, 32
		LAFM1	—	16	H'FFFFE186	16
		DATA_01	—	16	H'FFFFE188	8, 16, 32
		DATA_23	—	16	H'FFFFE18A	8, 16
		DATA_45	—	16	H'FFFFE18C	8, 16, 32
		DATA_67	—	16	H'FFFFE18E	8, 16
		CONTROL1	—	16	H'FFFFE190	8, 16
		TIMESTAMP	—	16	H'FFFFE192	16
	MB[5].	CONTROL0_H	—	16	H'FFFFE1A0	16, 32
		CONTROL0_L	—	16	H'FFFFE1A2	16
		LAFM0	—	16	H'FFFFE1A4	16, 32
		LAFM1	—	16	H'FFFFE1A6	16
		DATA_01	—	16	H'FFFFE1A8	8, 16, 32
		DATA_23	—	16	H'FFFFE1AA	8, 16
		DATA_45	—	16	H'FFFFE1AC	8, 16, 32
		DATA_67	—	16	H'FFFFE1AE	8, 16
		CONTROL1	—	16	H'FFFFE1B0	8, 16
		TIMESTAMP	—	16	H'FFFFE1B2	16

Module	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1 (RCAN_C)	MB[6].	CONTROL0_H	—	16	H'FFFFE1C0	16, 32
		CONTROL0_L	—	16	H'FFFFE1C2	16
		LAFM0	—	16	H'FFFFE1C4	16, 32
		LAFM1	—	16	H'FFFFE1C6	16
		DATA_01	—	16	H'FFFFE1C8	8, 16, 32
		DATA_23	—	16	H'FFFFE1CA	8, 16
		DATA_45	—	16	H'FFFFE1CC	8, 16, 32
		DATA_67	—	16	H'FFFFE1CE	8, 16
		CONTROL1	—	16	H'FFFFE1D0	8, 16
		TIMESTAMP	—	16	H'FFFFE1D2	16
	MB[7].	CONTROL0_H	—	16	H'FFFFE1E0	16, 32
		CONTROL0_L	—	16	H'FFFFE1E2	16
		LAFM0	—	16	H'FFFFE1E4	16, 32
		LAFM1	—	16	H'FFFFE1E6	16
		DATA_01	—	16	H'FFFFE1E8	8, 16, 32
		DATA_23	—	16	H'FFFFE1EA	8, 16
		DATA_45	—	16	H'FFFFE1EC	8, 16, 32
		DATA_67	—	16	H'FFFFE1EE	8, 16
		CONTROL1	—	16	H'FFFFE1F0	8, 16
		TIMESTAMP	—	16	H'FFFFE1F2	16
	MB[8].	CONTROL0_H	—	16	H'FFFFE200	16, 32
		CONTROL0_L	—	16	H'FFFFE202	16
		LAFM0	—	16	H'FFFFE204	16, 32
		LAFM1	—	16	H'FFFFE206	16
		DATA_01	—	16	H'FFFFE208	8, 16, 32
		DATA_23	—	16	H'FFFFE20A	8, 16
		DATA_45	—	16	H'FFFFE20C	8, 16, 32
		DATA_67	—	16	H'FFFFE20E	8, 16
		CONTROL1	—	16	H'FFFFE210	8, 16
		TIMESTAMP	—	16	H'FFFFE212	16

Module	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1 (RCAN_C)	MB[9].	CONTROL0_H	—	16	H'FFFFE220	16, 32
		CONTROL0_L	—	16	H'FFFFE222	16
		LAFM0	—	16	H'FFFFE224	16, 32
		LAFM1	—	16	H'FFFFE226	16
		DATA_01	—	16	H'FFFFE228	8, 16, 32
		DATA_23	—	16	H'FFFFE22A	8, 16
		DATA_45	—	16	H'FFFFE22C	8, 16, 32
		DATA_67	—	16	H'FFFFE22E	8, 16
		CONTROL1	—	16	H'FFFFE230	8, 16
		TIMESTAMP	—	16	H'FFFFE232	16
	MB[10].	CONTROL0_H	—	16	H'FFFFE240	16, 32
		CONTROL0_L	—	16	H'FFFFE242	16
		LAFM0	—	16	H'FFFFE244	16, 32
		LAFM1	—	16	H'FFFFE246	16
		DATA_01	—	16	H'FFFFE248	8, 16, 32
		DATA_23	—	16	H'FFFFE24A	8, 16
		DATA_45	—	16	H'FFFFE24C	8, 16, 32
		DATA_67	—	16	H'FFFFE24E	8, 16
		CONTROL1	—	16	H'FFFFE250	8, 16
		TIMESTAMP	—	16	H'FFFFE252	16
	MB[11].	CONTROL0_H	—	16	H'FFFFE260	16, 32
		CONTROL0_L	—	16	H'FFFFE262	16
		LAFM0	—	16	H'FFFFE264	16, 32
		LAFM1	—	16	H'FFFFE266	16
		DATA_01	—	16	H'FFFFE268	8, 16, 32
		DATA_23	—	16	H'FFFFE26A	8, 16
		DATA_45	—	16	H'FFFFE26C	8, 16, 32
		DATA_67	—	16	H'FFFFE26E	8, 16
		CONTROL1	—	16	H'FFFFE270	8, 16
		TIMESTAMP	—	16	H'FFFFE272	16

Module	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1 (RCAN_C)	MB[12].	CONTROL0_H	—	16	H'FFFFE280	16, 32
		CONTROL0_L	—	16	H'FFFFE282	16
		LAFM0	—	16	H'FFFFE284	16, 32
		LAFM1	—	16	H'FFFFE286	16
		DATA_01	—	16	H'FFFFE288	8, 16, 32
		DATA_23	—	16	H'FFFFE28A	8, 16
		DATA_45	—	16	H'FFFFE28C	8, 16, 32
		DATA_67	—	16	H'FFFFE28E	8, 16
		CONTROL1	—	16	H'FFFFE290	8, 16
		TIMESTAMP	—	16	H'FFFFE292	16
	MB[13].	CONTROL0_H	—	16	H'FFFFE2A0	16, 32
		CONTROL0_L	—	16	H'FFFFE2A2	16
		LAFM0	—	16	H'FFFFE2A4	16, 32
		LAFM1	—	16	H'FFFFE2A6	16
		DATA_01	—	16	H'FFFFE2A8	8, 16, 32
		DATA_23	—	16	H'FFFFE2AA	8, 16
		DATA_45	—	16	H'FFFFE2AC	8, 16, 32
		DATA_67	—	16	H'FFFFE2AE	8, 16
		CONTROL1	—	16	H'FFFFE2B0	8, 16
		TIMESTAMP	—	16	H'FFFFE2B2	16
	MB[14].	CONTROL0_H	—	16	H'FFFFE2C0	16, 32
		CONTROL0_L	—	16	H'FFFFE2C2	16
		LAFM0	—	16	H'FFFFE2C4	16, 32
		LAFM1	—	16	H'FFFFE2C6	16
		DATA_01	—	16	H'FFFFE2C8	8, 16, 32
		DATA_23	—	16	H'FFFFE2CA	8, 16
		DATA_45	—	16	H'FFFFE2CC	8, 16, 32
		DATA_67	—	16	H'FFFFE2CE	8, 16
		CONTROL1	—	16	H'FFFFE2D0	8, 16
		TIMESTAMP	—	16	H'FFFFE2D2	16

Module	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1 (RCAN_C)	MB[15].	CONTROL0_H	—	16	H'FFFFE2E0	16, 32
		CONTROL0_L	—	16	H'FFFFE2E2	16
		LAFM0	—	16	H'FFFFE2E4	16, 32
		LAFM1	—	16	H'FFFFE2E6	16
		DATA_01	—	16	H'FFFFE2E8	8, 16, 32
		DATA_23	—	16	H'FFFFE2EA	8, 16
		DATA_45	—	16	H'FFFFE2EC	8, 16, 32
		DATA_67	—	16	H'FFFFE2EE	8, 16
		CONTROL1	—	16	H'FFFFE2F0	8, 16
		TIMESTAMP	—	16	H'FFFFE2F2	16
	MB[16].	CONTROL0_H	—	16	H'FFFFE300	16, 32
		CONTROL0_L	—	16	H'FFFFE302	16
		LAFM0	—	16	H'FFFFE304	16, 32
		LAFM1	—	16	H'FFFFE306	16
		DATA_01	—	16	H'FFFFE308	8, 16, 32
		DATA_23	—	16	H'FFFFE30A	8, 16
		DATA_45	—	16	H'FFFFE30C	8, 16, 32
		DATA_67	—	16	H'FFFFE30E	8, 16
		CONTROL1	—	16	H'FFFFE310	8, 16
	MB[17].	CONTROL0_H	—	16	H'FFFFE320	16, 32
		CONTROL0_L	—	16	H'FFFFE322	16
		LAFM0	—	16	H'FFFFE324	16, 32
		LAFM1	—	16	H'FFFFE326	16
		DATA_01	—	16	H'FFFFE328	8, 16, 32
		DATA_23	—	16	H'FFFFE32A	8, 16
		DATA_45	—	16	H'FFFFE32C	8, 16, 32
		DATA_67	—	16	H'FFFFE32E	8, 16
		CONTROL1	—	16	H'FFFFE330	8, 16
	MB[18].	CONTROL0_H	—	16	H'FFFFE340	16, 32
		CONTROL0_L	—	16	H'FFFFE342	16
		LAFM0	—	16	H'FFFFE344	16, 32

Module	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1 (RCAN_C)	MB[18].	LAFM1	—	16	H'FFFFE346	16
		DATA_01	—	16	H'FFFFE348	8, 16, 32
		DATA_23	—	16	H'FFFFE34A	8, 16
		DATA_45	—	16	H'FFFFE34C	8, 16, 32
		DATA_67	—	16	H'FFFFE34E	8, 16
		CONTROL1	—	16	H'FFFFE350	8, 16
	MB[19].	CONTROL0_H	—	16	H'FFFFE360	16, 32
		CONTROL0_L	—	16	H'FFFFE362	16
		LAFM0	—	16	H'FFFFE364	16, 32
		LAFM1	—	16	H'FFFFE366	16
		DATA_01	—	16	H'FFFFE368	8, 16, 32
		DATA_23	—	16	H'FFFFE36A	8, 16
		DATA_45	—	16	H'FFFFE36C	8, 16, 32
		DATA_67	—	16	H'FFFFE36E	8, 16
		CONTROL1	—	16	H'FFFFE370	8, 16
	MB[20].	CONTROL0_H	—	16	H'FFFFE380	16, 32
		CONTROL0_L	—	16	H'FFFFE382	16
		LAFM0	—	16	H'FFFFE384	16, 32
		LAFM1	—	16	H'FFFFE386	16
		DATA_01	—	16	H'FFFFE388	8, 16, 32
		DATA_23	—	16	H'FFFFE38A	8, 16
		DATA_45	—	16	H'FFFFE38C	8, 16, 32
		DATA_67	—	16	H'FFFFE38E	8, 16
		CONTROL1	—	16	H'FFFFE390	8, 16
	MB[21].	CONTROL0_H	—	16	H'FFFFE3A0	16, 32
		CONTROL0_L	—	16	H'FFFFE3A2	16
		LAFM0	—	16	H'FFFFE3A4	16, 32
		LAFM1	—	16	H'FFFFE3A6	16
		DATA_01	—	16	H'FFFFE3A8	8, 16, 32
		DATA_23	—	16	H'FFFFE3AA	8, 16
		DATA_45	—	16	H'FFFFE3AC	8, 16, 32

Module	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1 (RCAN_C)	MB[21].	DATA_67	—	16	H'FFFFE3AE	8, 16
		CONTROL1	—	16	H'FFFFE3B0	8, 16
	MB[22].	CONTROL0_H	—	16	H'FFFFE3C0	16, 32
		CONTROL0_L	—	16	H'FFFFE3C2	16
		LAFM0	—	16	H'FFFFE3C4	16, 32
		LAFM1	—	16	H'FFFFE3C6	16
		DATA_01	—	16	H'FFFFE3C8	8, 16, 32
		DATA_23	—	16	H'FFFFE3CA	8, 16
		DATA_45	—	16	H'FFFFE3CC	8, 16, 32
		DATA_67	—	16	H'FFFFE3CE	8, 16
		CONTROL1	—	16	H'FFFFE3D0	8, 16
	MB[23].	CONTROL0_H	—	16	H'FFFFE3E0	16, 32
		CONTROL0_L	—	16	H'FFFFE3E2	16
		LAFM0	—	16	H'FFFFE3E4	16, 32
		LAFM1	—	16	H'FFFFE3E6	16
		DATA_01	—	16	H'FFFFE3E8	8, 16, 32
		DATA_23	—	16	H'FFFFE3EA	8, 16
		DATA_45	—	16	H'FFFFE3EC	8, 16, 32
		DATA_67	—	16	H'FFFFE3EE	8, 16
		CONTROL1	—	16	H'FFFFE3F0	8, 16
	MB[24].	CONTROL0_H	—	16	H'FFFFE400	16, 32
		CONTROL0_L	—	16	H'FFFFE402	16
		LAFM0	—	16	H'FFFFE404	16, 32
		LAFM1	—	16	H'FFFFE406	16
		DATA_01	—	16	H'FFFFE408	8, 16, 32
		DATA_23	—	16	H'FFFFE40A	8, 16
		DATA_45	—	16	H'FFFFE40C	8, 16, 32
		DATA_67	—	16	H'FFFFE40E	8, 16
		CONTROL1	—	16	H'FFFFE410	8, 16
		TTT	—	16	H'FFFFE414	16
		TTCONTROL	—	16	H'FFFFE416	16

Module	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1 (RCAN_C)	MB[25].	CONTROL0_H	—	16	H'FFFFE420	16, 32
		CONTROL0_L	—	16	H'FFFFE422	16
		LAFM0	—	16	H'FFFFE424	16, 32
		LAFM1	—	16	H'FFFFE426	16
		DATA_01	—	16	H'FFFFE428	8, 16, 32
		DATA_23	—	16	H'FFFFE42A	8, 16
		DATA_45	—	16	H'FFFFE42C	8, 16, 32
		DATA_67	—	16	H'FFFFE42E	8, 16
		CONTROL1	—	16	H'FFFFE430	8, 16
		TTT	—	16	H'FFFFE434	16
		TTCONTROL	—	16	H'FFFFE436	16
	MB[26].	CONTROL0_H	—	16	H'FFFFE440	16, 32
		CONTROL0_L	—	16	H'FFFFE442	16
		LAFM0	—	16	H'FFFFE444	16, 32
		LAFM1	—	16	H'FFFFE446	16
		DATA_01	—	16	H'FFFFE448	8, 16, 32
		DATA_23	—	16	H'FFFFE44A	8, 16
		DATA_45	—	16	H'FFFFE44C	8, 16, 32
		DATA_67	—	16	H'FFFFE44E	8, 16
		CONTROL1	—	16	H'FFFFE450	8, 16
		TTT	—	16	H'FFFFE454	16
		TTCONTROL	—	16	H'FFFFE456	16
	MB[27].	CONTROL0_H	—	16	H'FFFFE460	16, 32
		CONTROL0_L	—	16	H'FFFFE462	16
		LAFM0	—	16	H'FFFFE464	16, 32
		LAFM1	—	16	H'FFFFE466	16
		DATA_01	—	16	H'FFFFE468	8, 16, 32
		DATA_23	—	16	H'FFFFE46A	8, 16
		DATA_45	—	16	H'FFFFE46C	8, 16, 32
		DATA_67	—	16	H'FFFFE46E	8, 16

Module	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1 (RCAN_C)	MB[27].	CONTROL1	—	16	H'FFFFE470	8, 16
		TTT	—	16	H'FFFFE474	16
		TTCONTROL	—	16	H'FFFFE476	16
	MB[28].	CONTROL0_H	—	16	H'FFFFE480	16, 32
		CONTROL0_L	—	16	H'FFFFE482	16
		LAFM0	—	16	H'FFFFE484	16, 32
		LAFM1	—	16	H'FFFFE486	16
		DATA_01	—	16	H'FFFFE488	8, 16, 32
		DATA_23	—	16	H'FFFFE48A	8, 16
		DATA_45	—	16	H'FFFFE48C	8, 16, 32
		DATA_67	—	16	H'FFFFE48E	8, 16
		CONTROL1	—	16	H'FFFFE490	8, 16
		TTT	—	16	H'FFFFE494	16
		TTCONTROL	—	16	H'FFFFE496	16
	MB[29].	CONTROL0_H	—	16	H'FFFFE4A0	16, 32
		CONTROL0_L	—	16	H'FFFFE4A2	16
		LAFM0	—	16	H'FFFFE4A4	16, 32
		LAFM1	—	16	H'FFFFE4A6	16
		DATA_01	—	16	H'FFFFE4A8	8, 16, 32
		DATA_23	—	16	H'FFFFE4AA	8, 16
		DATA_45	—	16	H'FFFFE4AC	8, 16, 32
		DATA_67	—	16	H'FFFFE4AE	8, 16
		CONTROL1	—	16	H'FFFFE4B0	8, 16
		TTT	—	16	H'FFFFE4B4	16
		TTCONTROL	—	16	H'FFFFE4B6	16
	MB[30].	CONTROL0_H	—	16	H'FFFFE4C0	16, 32
		CONTROL0_L	—	16	H'FFFFE4C2	16
		LAFM0	—	16	H'FFFFE4C4	16, 32
		LAFM1	—	16	H'FFFFE4C6	16
		DATA_01	—	16	H'FFFFE4C8	8, 16, 32

Module	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1 (RCAN_C)	MB[30].	DATA_23	—	16	H'FFFFE4CA	8, 16
		DATA_45	—	16	H'FFFFE4CC	8, 16, 32
		DATA_67	—	16	H'FFFFE4CE	8, 16
		CONTROL1	—	16	H'FFFFE4D0	8, 16
		TIMESTAMP	—	16	H'FFFFE4D2	16
		TTT	—	16	H'FFFFE4D4	16
	MB[31].	CONTROL0_H	—	16	H'FFFFE4E0	16, 32
		CONTROL0_L	—	16	H'FFFFE4E2	16
		LAFM0	—	16	H'FFFFE4E4	16, 32
		LAFM1	—	16	H'FFFFE4E6	16
		DATA_01	—	16	H'FFFFE4E8	8, 16, 32
		DATA_23	—	16	H'FFFFE4EA	8, 16
		DATA_45	—	16	H'FFFFE4EC	8, 16, 32
		DATA_67	—	16	H'FFFFE4EE	8, 16
		CONTROL1	—	16	H'FFFFE4F0	8, 16
		TIMESTAMP	—	16	H'FFFFE4F2	16
	Message buffer error status register		MBESR	16	H'FFFFE600	16
	Message buffer error control register		MBECR	16	H'FFFFE602	16
RCAN-TL1 (RCAN_D)	Master control register		MCR	16	H'FFFF0000	16
	General status register		GSR	16	H'FFFF0002	16
	Bit configuration register 1		BCR1	16	H'FFFF0004	16
	Bit configuration register 0		BCR0	16	H'FFFF0006	16
	Interrupt request register		IRR	16	H'FFFF0008	16
	Interrupt mask register		IMR	16	H'FFFF000A	16
	Error counter register		TEC/REC	16	H'FFFF000C	16
	Transmit pending register 1		TXPR1	16	H'FFFF0020	32
	Transmit pending register 0		TXPR0	16		
	Transmit cancel register 1		TXCR1	16	H'FFFF0028	16

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1 (RCAN_D)	Transmit cancel register 0	TXCR0	16	H'FFFF002A	16
	Transmit acknowledge register 1	TXACK1	16	H'FFFF0030	16
	Transmit acknowledge register 0	TXACK0	16	H'FFFF0032	16
	Abort acknowledge register 1	ABACK1	16	H'FFFF0038	16
	Abort acknowledge register 0	ABACK0	16	H'FFFF003A	16
	Data frame receive pending register 1	RXPR1	16	H'FFFF0040	16
	Data frame receive pending register 0	RXPR0	16	H'FFFF0042	16
	Remote frame receive pending register 1	RFPR1	16	H'FFFF0048	16
	Remote frame receive pending register 0	RFPR0	16	H'FFFF004A	16
	Mailbox interrupt mask register 1	MBIMR1	16	H'FFFF0050	16
	Mailbox interrupt mask register 0	MBIMR0	16	H'FFFF0052	16
	Unread message status register 1	UMSR1	16	H'FFFF0058	16
	Unread message status register 0	UMSR0	16	H'FFFF005A	16
	Timer trigger control register 0	TTCR0	16	H'FFFF0080	16
	Cycle maximum/tx-enable window register	CMAX_TEW	16	H'FFFF0084	16
	Reference trigger offset register	RFTROFF	16	H'FFFF0086	16
	Timer status register	TSR	16	H'FFFF0088	16
	Cycle counter register	CCR	16	H'FFFF008A	16
	Timer counter register	TCNTR	16	H'FFFF008C	16
	Cycle time register	CYCTR	16	H'FFFF0090	16
	Reference mark register	RFMK	16	H'FFFF0094	16
	Timer compare match register 0	TCMR0	16	H'FFFF0098	16
	Timer compare match register 1	TCMR1	16	H'FFFF009C	16
	Timer compare match register 2	TCMR2	16	H'FFFF00A0	16
	Tx-trigger time selection register	TTTSEL	16	H'FFFF00A4	16

Module	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1 (RCAN_D)	MB[0].	CONTROL0_H	—	16	H'FFFF0100	16, 32
		CONTROL0_L	—	16	H'FFFF0102	16
		LAFM0	—	16	H'FFFF0104	16, 32
		LAFM1	—	16	H'FFFF0106	16
		DATA_01	—	16	H'FFFF0108	8, 16, 32
		DATA_23	—	16	H'FFFF010A	8, 16
		DATA_45	—	16	H'FFFF010C	8, 16, 32
		DATA_67	—	16	H'FFFF010E	8, 16
		CONTROL1	—	16	H'FFFF0110	8, 16
		TIMESTAMP	—	16	H'FFFF0112	16
	MB[1].	CONTROL0_H	—	16	H'FFFF0120	16, 32
		CONTROL0_L	—	16	H'FFFF0122	16
		LAFM0	—	16	H'FFFF0124	16, 32
		LAFM1	—	16	H'FFFF0126	16
		DATA_01	—	16	H'FFFF0128	8, 16, 32
		DATA_23	—	16	H'FFFF012A	8, 16
		DATA_45	—	16	H'FFFF012C	8, 16, 32
		DATA_67	—	16	H'FFFF012E	8, 16
		CONTROL1	—	16	H'FFFF0130	8, 16
		TIMESTAMP	—	16	H'FFFF0132	16
	MB[2].	CONTROL0_H	—	16	H'FFFF0140	16, 32
		CONTROL0_L	—	16	H'FFFF0142	16
		LAFM0	—	16	H'FFFF0144	16, 32
		LAFM1	—	16	H'FFFF0146	16
		DATA_01	—	16	H'FFFF0148	8, 16, 32
		DATA_23	—	16	H'FFFF014A	8, 16
		DATA_45	—	16	H'FFFF014C	8, 16, 32
		DATA_67	—	16	H'FFFF014E	8, 16
		CONTROL1	—	16	H'FFFF0150	8, 16
		TIMESTAMP	—	16	H'FFFF0152	16

Module	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1 (RCAN_D)	MB[3].	CONTROL0_H	—	16	H'FFFF0160	16, 32
		CONTROL0_L	—	16	H'FFFF0162	16
		LAFM0	—	16	H'FFFF0164	16, 32
		LAFM1	—	16	H'FFFF0166	16
		DATA_01	—	16	H'FFFF0168	8, 16, 32
		DATA_23	—	16	H'FFFF016A	8, 16
		DATA_45	—	16	H'FFFF016C	8, 16, 32
		DATA_67	—	16	H'FFFF016E	8, 16
		CONTROL1	—	16	H'FFFF0170	8, 16
		TIMESTAMP	—	16	H'FFFF0172	16
	MB[4].	CONTROL0_H	—	16	H'FFFF0180	16, 32
		CONTROL0_L	—	16	H'FFFF0182	16
		LAFM0	—	16	H'FFFF0184	16, 32
		LAFM1	—	16	H'FFFF0186	16
		DATA_01	—	16	H'FFFF0188	8, 16, 32
		DATA_23	—	16	H'FFFF018A	8, 16
		DATA_45	—	16	H'FFFF018C	8, 16, 32
		DATA_67	—	16	H'FFFF018E	8, 16
		CONTROL1	—	16	H'FFFF0190	8, 16
		TIMESTAMP	—	16	H'FFFF0192	16
	MB[5].	CONTROL0_H	—	16	H'FFFF01A0	16, 32
		CONTROL0_L	—	16	H'FFFF01A2	16
		LAFM0	—	16	H'FFFF01A4	16, 32
		LAFM1	—	16	H'FFFF01A6	16
		DATA_01	—	16	H'FFFF01A8	8, 16, 32
		DATA_23	—	16	H'FFFF01AA	8, 16
		DATA_45	—	16	H'FFFF01AC	8, 16, 32
		DATA_67	—	16	H'FFFF01AE	8, 16
		CONTROL1	—	16	H'FFFF01B0	8, 16
		TIMESTAMP	—	16	H'FFFF01B2	16

Module	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1 (RCAN_D)	MB[6].	CONTROL0_H	—	16	H'FFFF01C0	16, 32
		CONTROL0_L	—	16	H'FFFF01C2	16
		LAFM0	—	16	H'FFFF01C4	16, 32
		LAFM1	—	16	H'FFFF01C6	16
		DATA_01	—	16	H'FFFF01C8	8, 16, 32
		DATA_23	—	16	H'FFFF01CA	8, 16
		DATA_45	—	16	H'FFFF01CC	8, 16, 32
		DATA_67	—	16	H'FFFF01CE	8, 16
		CONTROL1	—	16	H'FFFF01D0	8, 16
		TIMESTAMP	—	16	H'FFFF01D2	16
	MB[7].	CONTROL0_H	—	16	H'FFFF01E0	16, 32
		CONTROL0_L	—	16	H'FFFF01E2	16
		LAFM0	—	16	H'FFFF01E4	16, 32
		LAFM1	—	16	H'FFFF01E6	16
		DATA_01	—	16	H'FFFF01E8	8, 16, 32
		DATA_23	—	16	H'FFFF01EA	8, 16
		DATA_45	—	16	H'FFFF01EC	8, 16, 32
		DATA_67	—	16	H'FFFF01EE	8, 16
		CONTROL1	—	16	H'FFFF01F0	8, 16
		TIMESTAMP	—	16	H'FFFF01F2	16
	MB[8].	CONTROL0_H	—	16	H'FFFF0200	16, 32
		CONTROL0_L	—	16	H'FFFF0202	16
		LAFM0	—	16	H'FFFF0204	16, 32
		LAFM1	—	16	H'FFFF0206	16
		DATA_01	—	16	H'FFFF0208	8, 16, 32
		DATA_23	—	16	H'FFFF020A	8, 16
		DATA_45	—	16	H'FFFF020C	8, 16, 32
		DATA_67	—	16	H'FFFF020E	8, 16
		CONTROL1	—	16	H'FFFF0210	8, 16
		TIMESTAMP	—	16	H'FFFF0212	16

Module	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1 (RCAN_D)	MB[9].	CONTROL0_H	—	16	H'FFFF0220	16, 32
		CONTROL0_L	—	16	H'FFFF0222	16
		LAFM0	—	16	H'FFFF0224	16, 32
		LAFM1	—	16	H'FFFF0226	16
		DATA_01	—	16	H'FFFF0228	8, 16, 32
		DATA_23	—	16	H'FFFF022A	8, 16
		DATA_45	—	16	H'FFFF022C	8, 16, 32
		DATA_67	—	16	H'FFFF022E	8, 16
		CONTROL1	—	16	H'FFFF0230	8, 16
		TIMESTAMP	—	16	H'FFFF0232	16
	MB[10].	CONTROL0_H	—	16	H'FFFF0240	16, 32
		CONTROL0_L	—	16	H'FFFF0242	16
		LAFM0	—	16	H'FFFF0244	16, 32
		LAFM1	—	16	H'FFFF0246	16
		DATA_01	—	16	H'FFFF0248	8, 16, 32
		DATA_23	—	16	H'FFFF024A	8, 16
		DATA_45	—	16	H'FFFF024C	8, 16, 32
		DATA_67	—	16	H'FFFF024E	8, 16
		CONTROL1	—	16	H'FFFF0250	8, 16
		TIMESTAMP	—	16	H'FFFF0252	16
	MB[11].	CONTROL0_H	—	16	H'FFFF0260	16, 32
		CONTROL0_L	—	16	H'FFFF0262	16
		LAFM0	—	16	H'FFFF0264	16, 32
		LAFM1	—	16	H'FFFF0266	16
		DATA_01	—	16	H'FFFF0268	8, 16, 32
		DATA_23	—	16	H'FFFF026A	8, 16
		DATA_45	—	16	H'FFFF026C	8, 16, 32
		DATA_67	—	16	H'FFFF026E	8, 16
		CONTROL1	—	16	H'FFFF0270	8, 16
		TIMESTAMP	—	16	H'FFFF0272	16

Module	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1 (RCAN_D)	MB[12].	CONTROL0_H	—	16	H'FFFF0280	16, 32
		CONTROL0_L	—	16	H'FFFF0282	16
		LAFM0	—	16	H'FFFF0284	16, 32
		LAFM1	—	16	H'FFFF0286	16
		DATA_01	—	16	H'FFFF0288	8, 16, 32
		DATA_23	—	16	H'FFFF028A	8, 16
		DATA_45	—	16	H'FFFF028C	8, 16, 32
		DATA_67	—	16	H'FFFF028E	8, 16
		CONTROL1	—	16	H'FFFF0290	8, 16
		TIMESTAMP	—	16	H'FFFF0292	16
	MB[13].	CONTROL0_H	—	16	H'FFFF02A0	16, 32
		CONTROL0_L	—	16	H'FFFF02A2	16
		LAFM0	—	16	H'FFFF02A4	16, 32
		LAFM1	—	16	H'FFFF02A6	16
		DATA_01	—	16	H'FFFF02A8	8, 16, 32
		DATA_23	—	16	H'FFFF02AA	8, 16
		DATA_45	—	16	H'FFFF02AC	8, 16, 32
		DATA_67	—	16	H'FFFF02AE	8, 16
		CONTROL1	—	16	H'FFFF02B0	8, 16
		TIMESTAMP	—	16	H'FFFF02B2	16
	MB[14].	CONTROL0_H	—	16	H'FFFF02C0	16, 32
		CONTROL0_L	—	16	H'FFFF02C2	16
		LAFM0	—	16	H'FFFF02C4	16, 32
		LAFM1	—	16	H'FFFF02C6	16
		DATA_01	—	16	H'FFFF02C8	8, 16, 32
		DATA_23	—	16	H'FFFF02CA	8, 16
		DATA_45	—	16	H'FFFF02CC	8, 16, 32
		DATA_67	—	16	H'FFFF02CE	8, 16
		CONTROL1	—	16	H'FFFF02D0	8, 16
		TIMESTAMP	—	16	H'FFFF02D2	16

Module	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1 (RCAN_D)	MB[15].	CONTROL0_H	—	16	H'FFFF02E0	16, 32
		CONTROL0_L	—	16	H'FFFF02E2	16
		LAFM0	—	16	H'FFFF02E4	16, 32
		LAFM1	—	16	H'FFFF02E6	16
		DATA_01	—	16	H'FFFF02E8	8, 16, 32
		DATA_23	—	16	H'FFFF02EA	8, 16
		DATA_45	—	16	H'FFFF02EC	8, 16, 32
		DATA_67	—	16	H'FFFF02EE	8, 16
		CONTROL1	—	16	H'FFFF02F0	8, 16
		TIMESTAMP	—	16	H'FFFF02F2	16
	MB[16].	CONTROL0_H	—	16	H'FFFF0300	16, 32
		CONTROL0_L	—	16	H'FFFF0302	16
		LAFM0	—	16	H'FFFF0304	16, 32
		LAFM1	—	16	H'FFFF0306	16
		DATA_01	—	16	H'FFFF0308	8, 16, 32
		DATA_23	—	16	H'FFFF030A	8, 16
		DATA_45	—	16	H'FFFF030C	8, 16, 32
		DATA_67	—	16	H'FFFF030E	8, 16
		CONTROL1	—	16	H'FFFF0310	8, 16
	MB[17].	CONTROL0_H	—	16	H'FFFF0320	16, 32
		CONTROL0_L	—	16	H'FFFF0322	16
		LAFM0	—	16	H'FFFF0324	16, 32
		LAFM1	—	16	H'FFFF0326	16
		DATA_01	—	16	H'FFFF0328	8, 16, 32
		DATA_23	—	16	H'FFFF032A	8, 16
		DATA_45	—	16	H'FFFF032C	8, 16, 32
		DATA_67	—	16	H'FFFF032E	8, 16
		CONTROL1	—	16	H'FFFF0330	8, 16
	MB[18].	CONTROL0_H	—	16	H'FFFF0340	16, 32
		CONTROL0_L	—	16	H'FFFF0342	16

Module	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1 (RCAN_D)	MB[18].	LAFM0	—	16	H'FFFF0344	16, 32
		LAFM1	—	16	H'FFFF0346	16
		DATA_01	—	16	H'FFFF0348	8, 16, 32
		DATA_23	—	16	H'FFFF034A	8, 16
		DATA_45	—	16	H'FFFF034C	8, 16, 32
		DATA_67	—	16	H'FFFF034E	8, 16
		CONTROL1	—	16	H'FFFF0350	8, 16
	MB[19].	CONTROL0_H	—	16	H'FFFF0360	16, 32
		CONTROL0_L	—	16	H'FFFF0362	16
		LAFM0	—	16	H'FFFF0364	16, 32
		LAFM1	—	16	H'FFFF0366	16
		DATA_01	—	16	H'FFFF0368	8, 16, 32
		DATA_23	—	16	H'FFFF036A	8, 16
		DATA_45	—	16	H'FFFF036C	8, 16, 32
		DATA_67	—	16	H'FFFF036E	8, 16
		CONTROL1	—	16	H'FFFF0370	8, 16
	MB[20].	CONTROL0_H	—	16	H'FFFF0380	16, 32
		CONTROL0_L	—	16	H'FFFF0382	16
		LAFM0	—	16	H'FFFF0384	16, 32
		LAFM1	—	16	H'FFFF0386	16
		DATA_01	—	16	H'FFFF0388	8, 16, 32
		DATA_23	—	16	H'FFFF038A	8, 16
		DATA_45	—	16	H'FFFF038C	8, 16, 32
		DATA_67	—	16	H'FFFF038E	8, 16
		CONTROL1	—	16	H'FFFF0390	8, 16
	MB[21].	CONTROL0_H	—	16	H'FFFF03A0	16, 32
		CONTROL0_L	—	16	H'FFFF03A2	16
		LAFM0	—	16	H'FFFF03A4	16, 32
		LAFM1	—	16	H'FFFF03A6	16
		DATA_01	—	16	H'FFFF03A8	8, 16, 32

Module	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1 (RCAN_D)	MB[22].	DATA_23	—	16	H'FFFF03AA	8, 16
		DATA_45	—	16	H'FFFF03AC	8, 16, 32
		DATA_67	—	16	H'FFFF03AE	8, 16
		CONTROL1	—	16	H'FFFF03B0	8, 16
		CONTROL0_H	—	16	H'FFFF03C0	16, 32
		CONTROL0_L	—	16	H'FFFF03C2	16
		LAFM0	—	16	H'FFFF03C4	16, 32
		LAFM1	—	16	H'FFFF03C6	16
		DATA_01	—	16	H'FFFF03C8	8, 16, 32
		DATA_23	—	16	H'FFFF03CA	8, 16
		DATA_45	—	16	H'FFFF03CC	8, 16, 32
		DATA_67	—	16	H'FFFF03CE	8, 16
		CONTROL1	—	16	H'FFFF03D0	8, 16
	MB[23].	CONTROL0_H	—	16	H'FFFF03E0	16, 32
		CONTROL0_L	—	16	H'FFFF03E2	16
		LAFM0	—	16	H'FFFF03E4	16, 32
		LAFM1	—	16	H'FFFF03E6	16
		DATA_01	—	16	H'FFFF03E8	8, 16, 32
		DATA_23	—	16	H'FFFF03EA	8, 16
		DATA_45	—	16	H'FFFF03EC	8, 16, 32
		DATA_67	—	16	H'FFFF03EE	8, 16
		CONTROL1	—	16	H'FFFF03F0	8, 16
	MB[24].	CONTROL0_H	—	16	H'FFFF0400	16, 32
		CONTROL0_L	—	16	H'FFFF0402	16
		LAFM0	—	16	H'FFFF0404	16, 32
		LAFM1	—	16	H'FFFF0406	16
		DATA_01	—	16	H'FFFF0408	8, 16, 32
		DATA_23	—	16	H'FFFF040A	8, 16
		DATA_45	—	16	H'FFFF040C	8, 16, 32
		DATA_67	—	16	H'FFFF040E	8, 16

Module	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1 (RCAN_D)	MB[24].	CONTROL1	—	16	H'FFFF0410	8, 16
		TTT	—	16	H'FFFF0414	16
		TTCONTROL	—	16	H'FFFF0416	16
	MB[25].	CONTROL0_H	—	16	H'FFFF0420	16, 32
		CONTROL0_L	—	16	H'FFFF0422	16
		LAFM0	—	16	H'FFFF0424	16, 32
		LAFM1	—	16	H'FFFF0426	16
		DATA_01	—	16	H'FFFF0428	8, 16, 32
		DATA_23	—	16	H'FFFF042A	8, 16
		DATA_45	—	16	H'FFFF042C	8, 16, 32
		DATA_67	—	16	H'FFFF042E	8, 16
		CONTROL1	—	16	H'FFFF0430	8, 16
		TTT	—	16	H'FFFF0434	16
		TTCONTROL	—	16	H'FFFF0436	16
	MB[26].	CONTROL0_H	—	16	H'FFFF0440	16, 32
		CONTROL0_L	—	16	H'FFFF0442	16
		LAFM0	—	16	H'FFFF0444	16, 32
		LAFM1	—	16	H'FFFF0446	16
		DATA_01	—	16	H'FFFF0448	8, 16, 32
		DATA_23	—	16	H'FFFF044A	8, 16
		DATA_45	—	16	H'FFFF044C	8, 16, 32
		DATA_67	—	16	H'FFFF044E	8, 16
		CONTROL1	—	16	H'FFFF0450	8, 16
		TTT	—	16	H'FFFF0454	16
		TTCONTROL	—	16	H'FFFF0456	16
	MB[27].	CONTROL0_H	—	16	H'FFFF0460	16, 32
		CONTROL0_L	—	16	H'FFFF0462	16
		LAFM0	—	16	H'FFFF0464	16, 32
		LAFM1	—	16	H'FFFF0466	16
		DATA_01	—	16	H'FFFF0468	8, 16, 32

Module	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1 (RCAN_D)	MB[27].	DATA_23	—	16	H'FFFF046A	8, 16
		DATA_45	—	16	H'FFFF046C	8, 16, 32
		DATA_67	—	16	H'FFFF046E	8, 16
		CONTROL1	—	16	H'FFFF0470	8, 16
		TTT	—	16	H'FFFF0474	16
		TTCONTROL	—	16	H'FFFF0476	16
	MB[28].	CONTROL0_H	—	16	H'FFFF0480	16, 32
		CONTROL0_L	—	16	H'FFFF0482	16
		LAFM0	—	16	H'FFFF0484	16, 32
		LAFM1	—	16	H'FFFF0486	16
		DATA_01	—	16	H'FFFF0488	8, 16, 32
		DATA_23	—	16	H'FFFF048A	8, 16
		DATA_45	—	16	H'FFFF048C	8, 16, 32
		DATA_67	—	16	H'FFFF048E	8, 16
		CONTROL1	—	16	H'FFFF0490	8, 16
		TTT	—	16	H'FFFF0494	16
		TTCONTROL	—	16	H'FFFF0496	16
	MB[29].	CONTROL0_H	—	16	H'FFFF04A0	16, 32
		CONTROL0_L	—	16	H'FFFF04A2	16
		LAFM0	—	16	H'FFFF04A4	16, 32
		LAFM1	—	16	H'FFFF04A6	16
		DATA_01	—	16	H'FFFF04A8	8, 16, 32
		DATA_23	—	16	H'FFFF04AA	8, 16
		DATA_45	—	16	H'FFFF04AC	8, 16, 32
		DATA_67	—	16	H'FFFF04AE	8, 16
		CONTROL1	—	16	H'FFFF04B0	8, 16
		TTT	—	16	H'FFFF04B4	16
		TTCONTROL	—	16	H'FFFF04B6	16

Module	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1 (RCAN_D)	MB[30].	CONTROL0_H	—	16	H'FFFF04C0	16, 32
		CONTROL0_L	—	16	H'FFFF04C2	16
		LAFM0	—	16	H'FFFF04C4	16, 32
		LAFM1	—	16	H'FFFF04C6	16
		DATA_01	—	16	H'FFFF04C8	8, 16, 32
		DATA_23	—	16	H'FFFF04CA	8, 16
		DATA_45	—	16	H'FFFF04CC	8, 16, 32
		DATA_67	—	16	H'FFFF04CE	8, 16
		CONTROL1	—	16	H'FFFF04D0	8, 16
		TIMESTAMP	—	16	H'FFFF04D2	16
		TTT	—	16	H'FFFF04D4	16
	MB[31].	CONTROL0_H	—	16	H'FFFF04E0	16, 32
		CONTROL0_L	—	16	H'FFFF04E2	16
		LAFM0	—	16	H'FFFF04E4	16, 32
		LAFM1	—	16	H'FFFF04E6	16
		DATA_01	—	16	H'FFFF04E8	8, 16, 32
		DATA_23	—	16	H'FFFF04EA	8, 16
		DATA_45	—	16	H'FFFF04EC	8, 16, 32
		DATA_67	—	16	H'FFFF04EE	8, 16
		CONTROL1	—	16	H'FFFF04F0	8, 16
		TIMESTAMP	—	16	H'FFFF04F2	16
	Message buffer error status register		MBESR	16	H'FFFF0600	16
	Message buffer error control register		MBECR	16	H'FFFF0602	16
FlexRay	FlexRay operation control register		FXROC	8	H'FFFC2004	8, 16, 32
	FlexRay operation status register		FXROS	8	H'FFFC2005	8, 16, 32

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
FlexRay	FlexRay timer interrupt status register	FXRTISR	8	H'FFFC200C	8, 16, 32
	FlexRay timer interrupt enable register	FXRTIER	8	H'FFFC200D	8, 16, 32
	FlexRay lock register	FRLCK	32	H'FFFC201C	8, 16, 32
	FlexRay error interrupt register	FREIR	32	H'FFFC2020	8, 16, 32
	FlexRay status interrupt register	FRSIR	32	H'FFFC2024	8, 16, 32
	FlexRay error interrupt line select	FREILS	32	H'FFFC2028	8, 16, 32
	FlexRay status interrupt line select	FRSILS	32	H'FFFC202C	8, 16, 32
	FlexRay error interrupt enable set	FREIES	32	H'FFFC2030	8, 16, 32
	FlexRay error interrupt enable reset	FREIER	32	H'FFFC2034	8, 16, 32
	FlexRay status interrupt enable set	FRSIES	32	H'FFFC2038	8, 16, 32
	FlexRay status interrupt enable reset	FRSIER	32	H'FFFC203C	8, 16, 32
	FlexRay interrupt line enable	FRILE	32	H'FFFC2040	8, 16, 32
	FlexRay timer 0 configuration	FRT0C	32	H'FFFC2044	8, 16, 32
	FlexRay timer 1 configuration	FRT1C	32	H'FFFC2048	8, 16, 32
	FlexRay stop watch register 1	FRSTPW1	32	H'FFFC204C	8, 16, 32
	FlexRay stop watch register 2	FRSTPW2	32	H'FFFC2050	8, 16, 32
	FlexRay SUC configuration register 1	FRSUCC1	32	H'FFFC2080	8, 16, 32
	FlexRay SUC configuration register 2	FRSUCC2	32	H'FFFC2084	8, 16, 32
	FlexRay SUC configuration register 3	FRSUCC3	32	H'FFFC2088	8, 16, 32
	FlexRay NEM configuration register	FRNEMC	32	H'FFFC208C	8, 16, 32

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
FlexRay	FlexRay PRT configuration register 1	FRPRTC1	32	H'FFFC2090	8, 16, 32
	FlexRay PRT configuration register 2	FRPRTC2	32	H'FFFC2094	8, 16, 32
	FlexRay MHD configuration register	FRMHDC	32	H'FFFC2098	8, 16, 32
	FlexRay GTU configuration register 1	FRGTUC1	32	H'FFFC20A0	8, 16, 32
	FlexRay GTU configuration register 2	FRGTUC2	32	H'FFFC20A4	8, 16, 32
	FlexRay GTU configuration register 3	FRGTUC3	32	H'FFFC20A8	8, 16, 32
	FlexRay GTU configuration register 4	FRGTUC4	32	H'FFFC20AC	8, 16, 32
	FlexRay GTU configuration register 5	FRGTUC5	32	H'FFFC20B0	8, 16, 32
	FlexRay GTU configuration register 6	FRGTUC6	32	H'FFFC20B4	8, 16, 32
	FlexRay GTU configuration register 7	FRGTUC7	32	H'FFFC20B8	8, 16, 32
	FlexRay GTU configuration register 8	FRGTUC8	32	H'FFFC20BC	8, 16, 32
	FlexRay GTU configuration register 9	FRGTUC9	32	H'FFFC20C0	8, 16, 32
	FlexRay GTU configuration register 10	FRGTUC10	32	H'FFFC20C4	8, 16, 32
	FlexRay GTU configuration register 11	FRGTUC11	32	H'FFFC20C8	8, 16, 32
	FlexRay CC status vector	FRCCSV	32	H'FFFC2100	8, 16, 32
	FlexRay CC error vector	FRCCEV	32	H'FFFC2104	8, 16, 32
	FlexRay slot counter value	FRSCV	32	H'FFFC2110	8, 16, 32
	FlexRay macrotick and cycle counter value	FRMTCCV	32	H'FFFC2114	8, 16, 32
	FlexRay rate correction value	FRRCV	32	H'FFFC2118	8, 16, 32

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
FlexRay	FlexRay offset correction value	FROCV	32	H'FFFC211C	8, 16, 32
	FlexRay sync frame status	FRSFS	32	H'FFFC2120	8, 16, 32
	FlexRay symbol window and NIT status	FRSWNIT	32	H'FFFC2124	8, 16, 32
	FlexRay aggregated channel status	FRACS	32	H'FFFC2128	8, 16, 32
	FlexRay even sync ID 1	FRESID1	32	H'FFFC2130	8, 16, 32
	FlexRay even sync ID 2	FRESID2	32	H'FFFC2134	8, 16, 32
	FlexRay even sync ID 3	FRESID3	32	H'FFFC2138	8, 16, 32
	FlexRay even sync ID 4	FRESID4	32	H'FFFC213C	8, 16, 32
	FlexRay even sync ID 5	FRESID5	32	H'FFFC2140	8, 16, 32
	FlexRay even sync ID 6	FRESID6	32	H'FFFC2144	8, 16, 32
	FlexRay even sync ID 7	FRESID7	32	H'FFFC2148	8, 16, 32
	FlexRay even sync ID 8	FRESID8	32	H'FFFC214C	8, 16, 32
	FlexRay even sync ID 9	FRESID9	32	H'FFFC2150	8, 16, 32
	FlexRay even sync ID 10	FRESID10	32	H'FFFC2154	8, 16, 32
	FlexRay even sync ID 11	FRESID11	32	H'FFFC2158	8, 16, 32
	FlexRay even sync ID 12	FRESID12	32	H'FFFC215C	8, 16, 32
	FlexRay even sync ID 13	FRESID13	32	H'FFFC2160	8, 16, 32
	FlexRay even sync ID 14	FRESID14	32	H'FFFC2164	8, 16, 32
	FlexRay even sync ID 15	FRESID15	32	H'FFFC2168	8, 16, 32
	FlexRay odd sync ID 1	FROSID1	32	H'FFFC2170	8, 16, 32
	FlexRay odd sync ID 2	FROSID2	32	H'FFFC2174	8, 16, 32
	FlexRay odd sync ID 3	FROSID3	32	H'FFFC2178	8, 16, 32
	FlexRay odd sync ID 4	FROSID4	32	H'FFFC217C	8, 16, 32
	FlexRay odd sync ID 5	FROSID5	32	H'FFFC2180	8, 16, 32
	FlexRay odd sync ID 6	FROSID6	32	H'FFFC2184	8, 16, 32
	FlexRay odd sync ID 7	FROSID7	32	H'FFFC2188	8, 16, 32
	FlexRay odd sync ID 8	FROSID8	32	H'FFFC218C	8, 16, 32

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
FlexRay	FlexRay odd sync ID 9	FROSID9	32	H'FFFC2190	8, 16, 32
	FlexRay odd sync ID 10	FROSID10	32	H'FFFC2194	8, 16, 32
	FlexRay odd sync ID 11	FROSID11	32	H'FFFC2198	8, 16, 32
	FlexRay odd sync ID 12	FROSID12	32	H'FFFC219C	8, 16, 32
	FlexRay odd sync ID 13	FROSID13	32	H'FFFC21A0	8, 16, 32
	FlexRay odd sync ID 14	FROSID14	32	H'FFFC21A4	8, 16, 32
	FlexRay odd sync ID 15	FROSID15	32	H'FFFC21A8	8, 16, 32
	FlexRay network management vector 1	FRNMV1	32	H'FFFC21B0	8, 16, 32
	FlexRay network management vector 2	FRNMV2	32	H'FFFC21B4	8, 16, 32
	FlexRay network management vector 3	FRNMV3	32	H'FFFC21B8	8, 16, 32
	FlexRay message RAM configuration	FRMRC	32	H'FFFC2300	8, 16, 32
	FlexRay FIFO rejection filter	FRFRF	32	H'FFFC2304	8, 16, 32
	FlexRay FIFO rejection filter mask	FRFRFM	32	H'FFFC2308	8, 16, 32
	FlexRay FIFO critical level	FRFCL	32	H'FFFC230C	8, 16, 32
	FlexRay message handler status	FRMHDS	32	H'FFFC2310	8, 16, 32
	FlexRay last dynamic transmit slot	FRLDTS	32	H'FFFC2314	8, 16, 32
	FlexRay FIFO status register	FRFSR	32	H'FFFC2318	8, 16, 32
	FlexRay message handler constraints flags	FRMHDF	32	H'FFFC231C	8, 16, 32
	FlexRay transmission request 1	FRTXRQ1	32	H'FFFC2320	8, 16, 32
	FlexRay transmission request 2	FRTXRQ2	32	H'FFFC2324	8, 16, 32
	FlexRay transmission request 3	FRTXRQ3	32	H'FFFC2328	8, 16, 32
	FlexRay transmission request 4	FRTXRQ4	32	H'FFFC232C	8, 16, 32
	FlexRay new data 1	FRNDAT1	32	H'FFFC2330	8, 16, 32
	FlexRay new data 2	FRNDAT2	32	H'FFFC2334	8, 16, 32
	FlexRay new data 3	FRNDAT3	32	H'FFFC2338	8, 16, 32
	FlexRay new data 4	FRNDAT4	32	H'FFFC233C	8, 16, 32

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
FlexRay	FlexRay message buffer status changed 1	FRMBSC1	32	H'FFFC2340	8, 16, 32
	FlexRay message buffer status changed 2	FRMBSC2	32	H'FFFC2344	8, 16, 32
	FlexRay message buffer status changed 3	FRMBSC3	32	H'FFFC2348	8, 16, 32
	FlexRay message buffer status changed 4	FRMBSC4	32	H'FFFC234C	8, 16, 32
	FlexRay write data section 1	FRWRDS1	32	H'FFFC2400	8, 16, 32
	FlexRay write data section 2	FRWRDS2	32	H'FFFC2404	8, 16, 32
	FlexRay write data section 3	FRWRDS3	32	H'FFFC2408	8, 16, 32
	FlexRay write data section 4	FRWRDS4	32	H'FFFC240C	8, 16, 32
	FlexRay write data section 5	FRWRDS5	32	H'FFFC2410	8, 16, 32
	FlexRay write data section 6	FRWRDS6	32	H'FFFC2414	8, 16, 32
	FlexRay write data section 7	FRWRDS7	32	H'FFFC2418	8, 16, 32
	FlexRay write data section 8	FRWRDS8	32	H'FFFC241C	8, 16, 32
	FlexRay write data section 9	FRWRDS9	32	H'FFFC2420	8, 16, 32
	FlexRay write data section 10	FRWRDS10	32	H'FFFC2424	8, 16, 32
	FlexRay write data section 11	FRWRDS11	32	H'FFFC2428	8, 16, 32
	FlexRay write data section 12	FRWRDS12	32	H'FFFC242C	8, 16, 32
	FlexRay write data section 13	FRWRDS13	32	H'FFFC2430	8, 16, 32
	FlexRay write data section 14	FRWRDS14	32	H'FFFC2434	8, 16, 32
	FlexRay write data section 15	FRWRDS15	32	H'FFFC2438	8, 16, 32
	FlexRay write data section 16	FRWRDS16	32	H'FFFC243C	8, 16, 32
	FlexRay write data section 17	FRWRDS17	32	H'FFFC2440	8, 16, 32
	FlexRay write data section 18	FRWRDS18	32	H'FFFC2444	8, 16, 32
	FlexRay write data section 19	FRWRDS19	32	H'FFFC2448	8, 16, 32
	FlexRay write data section 20	FRWRDS20	32	H'FFFC244C	8, 16, 32
	FlexRay write data section 21	FRWRDS21	32	H'FFFC2450	8, 16, 32
	FlexRay write data section 22	FRWRDS22	32	H'FFFC2454	8, 16, 32
	FlexRay write data section 23	FRWRDS23	32	H'FFFC2458	8, 16, 32

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
FlexRay	FlexRay write data section 24	FRWRDS24	32	H'FFFC245C	8, 16, 32
	FlexRay write data section 25	FRWRDS25	32	H'FFFC2460	8, 16, 32
	FlexRay write data section 26	FRWRDS26	32	H'FFFC2464	8, 16, 32
	FlexRay write data section 27	FRWRDS27	32	H'FFFC2468	8, 16, 32
	FlexRay write data section 28	FRWRDS28	32	H'FFFC246C	8, 16, 32
	FlexRay write data section 29	FRWRDS29	32	H'FFFC2470	8, 16, 32
	FlexRay write data section 30	FRWRDS30	32	H'FFFC2474	8, 16, 32
	FlexRay write data section 31	FRWRDS31	32	H'FFFC2478	8, 16, 32
	FlexRay write data section 32	FRWRDS32	32	H'FFFC247C	8, 16, 32
	FlexRay write data section 33	FRWRDS33	32	H'FFFC2480	8, 16, 32
	FlexRay write data section 34	FRWRDS34	32	H'FFFC2484	8, 16, 32
	FlexRay write data section 35	FRWRDS35	32	H'FFFC2488	8, 16, 32
	FlexRay write data section 36	FRWRDS36	32	H'FFFC248C	8, 16, 32
	FlexRay write data section 37	FRWRDS37	32	H'FFFC2490	8, 16, 32
	FlexRay write data section 38	FRWRDS38	32	H'FFFC2494	8, 16, 32
	FlexRay write data section 39	FRWRDS39	32	H'FFFC2498	8, 16, 32
	FlexRay write data section 40	FRWRDS40	32	H'FFFC249C	8, 16, 32
	FlexRay write data section 41	FRWRDS41	32	H'FFFC24A0	8, 16, 32
	FlexRay write data section 42	FRWRDS42	32	H'FFFC24A4	8, 16, 32
	FlexRay write data section 43	FRWRDS43	32	H'FFFC24A8	8, 16, 32
	FlexRay write data section 44	FRWRDS44	32	H'FFFC24AC	8, 16, 32
	FlexRay write data section 45	FRWRDS45	32	H'FFFC24B0	8, 16, 32
	FlexRay write data section 46	FRWRDS46	32	H'FFFC24B4	8, 16, 32
	FlexRay write data section 47	FRWRDS47	32	H'FFFC24B8	8, 16, 32
	FlexRay write data section 48	FRWRDS48	32	H'FFFC24BC	8, 16, 32
	FlexRay write data section 49	FRWRDS49	32	H'FFFC24C0	8, 16, 32
	FlexRay write data section 50	FRWRDS50	32	H'FFFC24C4	8, 16, 32
	FlexRay write data section 51	FRWRDS51	32	H'FFFC24C8	8, 16, 32
	FlexRay write data section 52	FRWRDS52	32	H'FFFC24CC	8, 16, 32
	FlexRay write data section 53	FRWRDS53	32	H'FFFC24D0	8, 16, 32

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
FlexRay	FlexRay write data section 54	FRWRDS54	32	H'FFFC24D4	8, 16, 32
	FlexRay write data section 55	FRWRDS55	32	H'FFFC24D8	8, 16, 32
	FlexRay write data section 56	FRWRDS56	32	H'FFFC24DC	8, 16, 32
	FlexRay write data section 57	FRWRDS57	32	H'FFFC24E0	8, 16, 32
	FlexRay write data section 58	FRWRDS58	32	H'FFFC24E4	8, 16, 32
	FlexRay write data section 59	FRWRDS59	32	H'FFFC24E8	8, 16, 32
	FlexRay write data section 60	FRWRDS60	32	H'FFFC24EC	8, 16, 32
	FlexRay write data section 61	FRWRDS61	32	H'FFFC24F0	8, 16, 32
	FlexRay write data section 62	FRWRDS62	32	H'FFFC24F4	8, 16, 32
	FlexRay write data section 63	FRWRDS63	32	H'FFFC24F8	8, 16, 32
	FlexRay write data section 64	FRWRDS64	32	H'FFFC24FC	8, 16, 32
	FlexRay write header section 1	FRWRHS1	32	H'FFFC2500	8, 16, 32
	FlexRay write header section 2	FRWRHS2	32	H'FFFC2504	8, 16, 32
	FlexRay write header section 3	FRWRHS3	32	H'FFFC2508	8, 16, 32
	FlexRay input buffer command mask	FRIBCM	32	H'FFFC2510	8, 16, 32
	FlexRay input buffer command request	FRIBCR	32	H'FFFC2514	8, 16, 32
	FlexRay read data section 1	FRRDDS1	32	H'FFFC2600	8, 16, 32
	FlexRay read data section 2	FRRDDS2	32	H'FFFC2604	8, 16, 32
	FlexRay read data section 3	FRRDDS3	32	H'FFFC2608	8, 16, 32
	FlexRay read data section 4	FRRDDS4	32	H'FFFC260C	8, 16, 32
	FlexRay read data section 5	FRRDDS5	32	H'FFFC2610	8, 16, 32
	FlexRay read data section 6	FRRDDS6	32	H'FFFC2614	8, 16, 32
	FlexRay read data section 7	FRRDDS7	32	H'FFFC2618	8, 16, 32
	FlexRay read data section 8	FRRDDS8	32	H'FFFC261C	8, 16, 32
	FlexRay read data section 9	FRRDDS9	32	H'FFFC2620	8, 16, 32
	FlexRay read data section 10	FRRDDS10	32	H'FFFC2624	8, 16, 32
	FlexRay read data section 11	FRRDDS11	32	H'FFFC2628	8, 16, 32
	FlexRay read data section 12	FRRDDS12	32	H'FFFC262C	8, 16, 32

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
FlexRay	FlexRay read data section 13	FRRDDS13	32	H'FFFC2630	8, 16, 32
	FlexRay read data section 14	FRRDDS14	32	H'FFFC2634	8, 16, 32
	FlexRay read data section 15	FRRDDS15	32	H'FFFC2638	8, 16, 32
	FlexRay read data section 16	FRRDDS16	32	H'FFFC263C	8, 16, 32
	FlexRay read data section 17	FRRDDS17	32	H'FFFC2640	8, 16, 32
	FlexRay read data section 18	FRRDDS18	32	H'FFFC2644	8, 16, 32
	FlexRay read data section 19	FRRDDS19	32	H'FFFC2648	8, 16, 32
	FlexRay read data section 20	FRRDDS20	32	H'FFFC264C	8, 16, 32
	FlexRay read data section 21	FRRDDS21	32	H'FFFC2650	8, 16, 32
	FlexRay read data section 22	FRRDDS22	32	H'FFFC2654	8, 16, 32
	FlexRay read data section 23	FRRDDS23	32	H'FFFC2658	8, 16, 32
	FlexRay read data section 24	FRRDDS24	32	H'FFFC265C	8, 16, 32
	FlexRay read data section 25	FRRDDS25	32	H'FFFC2660	8, 16, 32
	FlexRay read data section 26	FRRDDS26	32	H'FFFC2664	8, 16, 32
	FlexRay read data section 27	FRRDDS27	32	H'FFFC2668	8, 16, 32
	FlexRay read data section 28	FRRDDS28	32	H'FFFC266C	8, 16, 32
	FlexRay read data section 29	FRRDDS29	32	H'FFFC2670	8, 16, 32
	FlexRay read data section 30	FRRDDS30	32	H'FFFC2674	8, 16, 32
	FlexRay read data section 31	FRRDDS31	32	H'FFFC2678	8, 16, 32
	FlexRay read data section 32	FRRDDS32	32	H'FFFC267C	8, 16, 32
	FlexRay read data section 33	FRRDDS33	32	H'FFFC2680	8, 16, 32
	FlexRay read data section 34	FRRDDS34	32	H'FFFC2684	8, 16, 32
	FlexRay read data section 35	FRRDDS35	32	H'FFFC2688	8, 16, 32
	FlexRay read data section 36	FRRDDS36	32	H'FFFC268C	8, 16, 32
	FlexRay read data section 37	FRRDDS37	32	H'FFFC2690	8, 16, 32
	FlexRay read data section 38	FRRDDS38	32	H'FFFC2694	8, 16, 32
	FlexRay read data section 39	FRRDDS39	32	H'FFFC2698	8, 16, 32
	FlexRay read data section 40	FRRDDS40	32	H'FFFC269C	8, 16, 32
	FlexRay read data section 41	FRRDDS41	32	H'FFFC26A0	8, 16, 32
	FlexRay read data section 42	FRRDDS42	32	H'FFFC26A4	8, 16, 32

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
FlexRay	FlexRay read data section 43	FRRDDS43	32	H'FFFC26A8	8, 16, 32
	FlexRay read data section 44	FRRDDS44	32	H'FFFC26AC	8, 16, 32
	FlexRay read data section 45	FRRDDS45	32	H'FFFC26B0	8, 16, 32
	FlexRay read data section 46	FRRDDS46	32	H'FFFC26B4	8, 16, 32
	FlexRay read data section 47	FRRDDS47	32	H'FFFC26B8	8, 16, 32
	FlexRay read data section 48	FRRDDS48	32	H'FFFC26BC	8, 16, 32
	FlexRay read data section 49	FRRDDS49	32	H'FFFC26C0	8, 16, 32
	FlexRay read data section 50	FRRDDS50	32	H'FFFC26C4	8, 16, 32
	FlexRay read data section 51	FRRDDS51	32	H'FFFC26C8	8, 16, 32
	FlexRay read data section 52	FRRDDS52	32	H'FFFC26CC	8, 16, 32
	FlexRay read data section 53	FRRDDS53	32	H'FFFC26D0	8, 16, 32
	FlexRay read data section 54	FRRDDS54	32	H'FFFC26D4	8, 16, 32
	FlexRay read data section 55	FRRDDS55	32	H'FFFC26D8	8, 16, 32
	FlexRay read data section 56	FRRDDS56	32	H'FFFC26DC	8, 16, 32
	FlexRay read data section 57	FRRDDS57	32	H'FFFC26E0	8, 16, 32
	FlexRay read data section 58	FRRDDS58	32	H'FFFC26E4	8, 16, 32
	FlexRay read data section 59	FRRDDS59	32	H'FFFC26E8	8, 16, 32
	FlexRay read data section 60	FRRDDS60	32	H'FFFC26EC	8, 16, 32
	FlexRay read data section 61	FRRDDS61	32	H'FFFC26F0	8, 16, 32
	FlexRay read data section 62	FRRDDS62	32	H'FFFC26F4	8, 16, 32
	FlexRay read data section 63	FRRDDS63	32	H'FFFC26F8	8, 16, 32
	FlexRay read data section 64	FRRDDS64	32	H'FFFC26FC	8, 16, 32
	FlexRay read header section 1	FRRDHS1	32	H'FFFC2700	8, 16, 32
	FlexRay read header section 2	FRRDHS2	32	H'FFFC2704	8, 16, 32
	FlexRay read header section 3	FRRDHS3	32	H'FFFC2708	8, 16, 32
	FlexRay message buffer status	FRMBS	32	H'FFFC270C	8, 16, 32
	FlexRay output buffer command mask	FROBCM	32	H'FFFC2710	8, 16, 32
	FlexRay output buffer command request	FROBCR	32	H'FFFC2714	8, 16, 32

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
ADC (ADC_A)	A/D data register Diag0	ADRD0	16	H'FFFFE83E	16
	A/D data register 0	ADR0	16	H'FFFFE840	16
	A/D data register 1	ADR1	16	H'FFFFE842	16
	A/D data register 2	ADR2	16	H'FFFFE844	16
	A/D data register 3	ADR3	16	H'FFFFE846	16
	A/D data register 4	ADR4	16	H'FFFFE848	16
	A/D data register 5	ADR5	16	H'FFFFE84A	16
	A/D data register 6	ADR6	16	H'FFFFE84C	16
	A/D data register 7	ADR7	16	H'FFFFE84E	16
	A/D data register 8	ADR8	16	H'FFFFE850	16
	A/D data register 9	ADR9	16	H'FFFFE852	16
	A/D data register 10	ADR10	16	H'FFFFE854	16
	A/D data register 11	ADR11	16	H'FFFFE856	16
	A/D data register 12	ADR12	16	H'FFFFE858	16
	A/D data register 13	ADR13	16	H'FFFFE85A	16
	A/D data register 14	ADR14	16	H'FFFFE85C	16
	A/D data register 15	ADR15	16	H'FFFFE85E	16
	A/D data register 16	ADR16	16	H'FFFFE860	16
	A/D data register 17	ADR17	16	H'FFFFE862	16
	A/D data register 18	ADR18	16	H'FFFFE864	16
	A/D data register 19	ADR19	16	H'FFFFE866	16
	A/D data register 20	ADR20	16	H'FFFFE868	16
	A/D data register 21	ADR21	16	H'FFFFE86A	16
	A/D data register 22	ADR22	16	H'FFFFE86C	16
	A/D data register 23	ADR23	16	H'FFFFE86E	16
	A/D data register 24	ADR24	16	H'FFFFE870	16
	A/D data register 25	ADR25	16	H'FFFFE872	16
	A/D data register 26	ADR26	16	H'FFFFE874	16
	A/D data register 27	ADR27	16	H'FFFFE876	16

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
ADC (ADC_B)	A/D data register Diag1	ADRD1	16	H'FFFFEC3E	16
	A/D data register 40	ADR40	16	H'FFFFEC40	16
	A/D data register 41	ADR41	16	H'FFFFEC42	16
	A/D data register 42	ADR42	16	H'FFFFEC44	16
	A/D data register 43	ADR43	16	H'FFFFEC46	16
	A/D data register 44	ADR44	16	H'FFFFEC48	16
	A/D data register 45	ADR45	16	H'FFFFEC4A	16
	A/D data register 46	ADR46	16	H'FFFFEC4C	16
	A/D data register 47	ADR47	16	H'FFFFEC4E	16
	A/D data register 48	ADR48	16	H'FFFFEC50	16
ADC (ADC_A)	A/D control register 0	ADCSR0	8	H'FFFFE800	8
ADC (ADC_B)	A/D control register 1	ADCSR1	8	H'FFFFEC00	8
ADC (ADC_A)	A/D conversion status register 0	ADREF0	8	H'FFFFE802	8
ADC (ADC_B)	A/D conversion status register 1	ADREF1	8	H'FFFFEC02	8
ADC (ADC_A)	A/D interrupt trigger enable register 0	ADTRE0	16	H'FFFFE804	8, 16
ADC (ADC_B)	A/D interrupt trigger enable register 1	ADTRE1	8	H'FFFFEC10	8
ADC (ADC_A)	A/D interrupt trigger conversion end flag register 0	ADTRF0	16	H'FFFFE806	8, 16
ADC (ADC_B)	A/D interrupt trigger conversion end flag register 1	ADTRF1	8	H'FFFFEC12	8
ADC (ADC_A)	A/D interrupt trigger source select register 0	ADTRS0	16	H'FFFFE808	8, 16
ADC (ADC_B)	A/D interrupt trigger source select register 1	ADTRS1	8	H'FFFFEC14	8
ADC (ADC_A)	A/D interrupt software trigger register 0	ADSTRG0	16	H'FFFFE80A	8, 16
ADC (ADC_B)	A/D interrupt software trigger register 1	ADSTRG1	8	H'FFFFEC16	8

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
ADC (ADC_A)	A/D interrupt trigger conversion end interrupt enable register 0	ADTRD0	16	H'FFFFE80C	8, 16
ADC (ADC_B)	A/D interrupt trigger conversion end interrupt enable register 1	ADTRD1	8	H'FFFFEC18	8
ADC (ADC_A)	A/D-converted value addition mode select register 0	ADADS0	8	H'FFFFE81C	8
	A/D-converted value addition mode select register 2	ADADS2	8	H'FFFFE924	8
	A/D-converted value addition mode select register 3	ADADS3	8	H'FFFFE926	8
	A/D-converted value addition mode select register 4	ADADS4	8	H'FFFFE928	8
ADC (ADC_B)	A/D-converted value addition mode select register 1	ADADS1	8	H'FFFFEC1C	8
	A/D-converted value addition mode select register 5	ADADS5	8	H'FFFFED24	8
ADC (ADC_A)	A/D-converted value addition count selection register 0	ADADC0	8	H'FFFFE81E	8
ADC (ADC_B)	A/D-converted value addition count selection register 1	ADADC1	8	H'FFFFEC1E	8
ADC (ADC_A)	A/D channel select register 0	ADANS0	16	H'FFFFE820	8, 16, 32
	A/D channel select register 1	ADANS1	16	H'FFFFE822	8, 16, 32
ADC (ADC_B)	A/D channel select register 3	ADANS3	16	H'FFFFEC20	8, 16
ADC (ADC_A)	A/D control extended register 0	ADCER0	16	H'FFFFE830	8, 16
ADC (ADC_B)	A/D control extended register 1	ADCER1	16	H'FFFFEC30	8, 16
JTAG	Instruction register	SDIR	4	—	—
	ID register	SDID	32	—	—
	Bypass register	SDBPR	1	—	—
	Boundary scan register	SDBSR	—	—	—

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
AUD-II	AUD control register	AUCSR	16	H'FFFC0C00	8, 16
	AUD window A start address register	AUWASR	32	H'FFFC0C04	8, 16, 32
	AUD window A end address register	AUWAER	32	H'FFFC0C08	8, 16, 32
	AUD window B start address register	AUWBSR	32	H'FFFC0C0C	8, 16, 32
	AUD window B end address register	AUWBER	32	H'FFFC0C10	8, 16, 32
	AUD extended control register	AUECSR	16	H'FFFC0C14	8, 16
PFC	Port A I/O register	PAIOR	16	H'FFFE3806	8, 16
	Port A control register 4	PACR4	16	H'FFFE3810	8, 16, 32
	Port A control register 3	PACR3	16	H'FFFE3812	8, 16
	Port A control register 2	PACR2	16	H'FFFE3814	8, 16, 32
	Port A control register 1	PACR1	16	H'FFFE3816	8, 16
	Port B I/O register	PBIOR	16	H'FFFE3886	8, 16
	Port B control register 4	PBCR4	16	H'FFFE3890	8, 16, 32
	Port B control register 3	PBCR3	16	H'FFFE3892	8, 16
	Port B control register 2	PBCR2	16	H'FFFE3894	8, 16, 32
	Port B control register 1	PBCR1	16	H'FFFE3896	8, 16
	Port C I/O register	PCIOR	16	H'FFFE3906	8, 16
	Port C control register 4	PCCR4	16	H'FFFE3910	8, 16, 32
	Port C control register 3	PCCR3	16	H'FFFE3912	8, 16
	Port C control register 2	PCCR2	16	H'FFFE3914	8, 16, 32
	Port C control register 1	PCCR1	16	H'FFFE3916	8, 16
	Port D I/O register	PDIOR	16	H'FFFFC808	8, 16
	Port D control register 2	PDCR2	16	H'FFFFC80C	8, 16, 32
	Port D control register 1	PDCR1	16	H'FFFFC80E	8, 16

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
PFC	Port E I/O register	PEIOR	16	H'FFFFC818	8, 16
	Port E control register 2	PECR2	16	H'FFFFC81C	8, 16, 32
	Port E control register 1	PECR1	16	H'FFFFC81E	8, 16
	Port F I/O register	PFIOR	16	H'FFFFC82A	8, 16
	Port F control register 2	PFCR2	16	H'FFFFC82C	8, 16, 32
	Port F control register 1	PFCR1	16	H'FFFFC82E	8, 16
	Port G I/O register	PGIOR	16	H'FFFFC83C	8, 16
	Port G control register 2	PGCR2	16	H'FFFFC840	8, 16, 32
	Port G control register 1	PGCR1	16	H'FFFFC842	8, 16
	Port H I/O register	PHIOR	16	H'FFFFC854	8, 16
	Port H control register	PHCR	16	H'FFFFC858	8, 16
	Port J I/O register	PJIOR	16	H'FFFFC86C	8, 16
	Port J control register 2	PJCR2	16	H'FFFFC870	8, 16, 32
	Port J control register 1	PJCR1	16	H'FFFFC872	8, 16
	Port K I/O register	PKIOR	16	H'FFFFC88C	8, 16
	Port K control register 2	PKCR2	16	H'FFFFC890	8, 16, 32
	Port K control register 1	PKCR1	16	H'FFFFC892	8, 16
	Port L I/O register	PLIOR	16	H'FFFFC8A8	8, 16
	Port L control register 2	PLCR2	16	H'FFFFC8AC	8, 16, 32
	Port L control register 1	PLCR1	16	H'FFFFC8AE	8, 16
	Port A control register 4A	PACR4A	16	H'FFFFCA00	8, 16, 32
	Port A control register 3A	PACR3A	16	H'FFFFCA02	8, 16
	Port A control register 2A	PACR2A	16	H'FFFFCA04	8, 16
	Port B control register 4A	PBCR4A	16	H'FFFFCA10	8, 16, 32
	Port B control register 3A	PBCR3A	16	H'FFFFCA12	8, 16
	Port B control register 2A	PBCR2A	16	H'FFFFCA14	8, 16, 32
	Port B control register 1A	PBCR1A	16	H'FFFFCA16	8, 16
	Port C control register 4A	PCCR4A	16	H'FFFFCA20	8, 16, 32

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
PFC	Port C control register 3A	PCCR3A	16	H'FFFFCA22	8, 16
	Port C control register 2A	PCCR2A	16	H'FFFFCA24	8, 16
	Port D control register 2A	PDCR2A	16	H'FFFFCA34	8, 16
	Port E control register 1A	PECR1A	16	H'FFFFCA46	8, 16
	Port F control register 2A	PFCR2A	16	H'FFFFCA54	8, 16
	Port J control register 3A	PJCR3A	16	H'FFFFCA82	8, 16
	Port J control register 2A	PJCR2A	16	H'FFFFCA84	8, 16, 32
	Port J control register 1A	PJCR1A	16	H'FFFFCA86	8, 16
I/O port	Port A data register	PADR	16	H'FFFE3802	8, 16
	Port A port register	PAPR	16	H'FFFE381E	8, 16
	Port A inverting register	PAIR	16	H'FFFE3818	8, 16
	Port B data register	PBDR	16	H'FFFE3882	8, 16
	Port B port register	PBPR	16	H'FFFE389E	8, 16
	Port B inverting register	PBIR	16	H'FFFE3898	8, 16, 32
	Port B driving ability setting register	PBDSR	16	H'FFFE389A	8, 16
	Port B pin state setting register	PBPSR	16	H'FFFE389C	8, 16, 32
	Port C data register	PCDR	16	H'FFFE3902	8, 16
	Port C port register	PCPR	16	H'FFFE391E	8, 16
	Port C inverting register	PCIR	16	H'FFFE3918	8, 16, 32
	Port C driving ability setting register	PCDSR	16	H'FFFE391A	8, 16
	Port C pin state setting register	PCPSR	16	H'FFFE391C	8, 16, 32
	Port D data register	PDDR	16	H'FFFFC800	8, 16, 32
	Port D port register	PDPR	16	H'FFFFC802	8, 16
	Port D inverting register	PDIR	16	H'FFFFC804	8, 16
	Port E data register	PEDR	16	H'FFFFC810	8, 16, 32
	Port E port register	PEPR	16	H'FFFFC812	8, 16
	Port E inverting register	PEIR	16	H'FFFFC814	8, 16, 32

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
I/O port	Port E driving ability setting register	PEDSR	16	H'FFFFC816	8, 16
	Port F data register	PFDR	16	H'FFFFC820	8, 16, 32
	Port F port register	PFPR	16	H'FFFFC822	8, 16
	Port F inverting register	PFIR	16	H'FFFFC824	8, 16, 32
	Port F driving ability setting register	PFDSR	16	H'FFFFC826	8, 16
	Port F pin state setting register	PFPSR	16	H'FFFFC828	8, 16, 32
	Port G data register	PGDR	16	H'FFFFC830	8, 16, 32
	Port G port register	PGPR	16	H'FFFFC832	8, 16
	Port G inverting register	PGIR	16	H'FFFFC834	8, 16, 32
	Port G driving ability setting register	PGDSR	16	H'FFFFC836	8, 16
	Port G edge selecting register	PGER	16	H'FFFFC838	8, 16
	Port H data register	PHDR	16	H'FFFFC850	8, 16, 32
	Port H port register	PHPR	16	H'FFFFC852	8, 16
	Port J data register	PJDR	16	H'FFFFC860	8, 16, 32
	Port J port register	PJPR	16	H'FFFFC862	8, 16
	Port J inverting register	PJIR	16	H'FFFFC864	8, 16, 32
	Port J driving ability setting register	PJDSR	16	H'FFFFC866	8, 16
	Port J pin state setting register	PJPSR	16	H'FFFFC868	8, 16
	Port K data register	PKDR	16	H'FFFFC880	8, 16, 32
	Port K port register	PKPR	16	H'FFFFC882	8, 16
	Port K inverting register	PKIR	16	H'FFFFC884	8, 16, 32
	Port K driving ability setting register	PKDSR	16	H'FFFFC886	8, 16
	Port K pin state setting register	PKPSR	16	H'FFFFC888	8, 16
	Port L data register	PLDR	16	H'FFFFC8A0	8, 16, 32
	Port L port register	PLPR	16	H'FFFFC8A2	8, 16
	Port L inverting register	PLIR	16	H'FFFFC8A4	8, 16
	CK control register	CKCR	16	H'FFFFC920	8, 16

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
MISG	Calculation data register	MISRCDR	32	H'FFF7FFFC	8, 16, 32
	Multi-input signature register	MISR	32	H'FFFC1C04	32
	Multi-input signature register 2	MISR2	32	H'FFFC1C08	32
	MISR control register	MISRCR	8	H'FFFC1C00	8
ROM/ IFS/ EEPROM	Flash pin monitor register	FPMON	8	H'FFFFA800	8
	Flash mode register	FMODR	8	H'FFFFA802	8
	Flash access status register	FASTAT	8	H'FFFFA810	8
	Flash access error interrupt enable register	FAEINT	8	H'FFFFA811	8
	ROM MAT select register	ROMMAT	16	H'FFFFA820	8, 16
	EEPROM read enable register 0	EEPRE0	16	H'FFFFA840	8, 16
	EEPROM read enable register 1	EEPRE1	16	H'FFFFA842	8, 16
	EEPROM program/erase enable register 0	EEPWE0	16	H'FFFFA850	8, 16
	EEPROM program/erase enable register 1	EEPWE1	16	H'FFFFA852	8, 16
	FCU RAM enable register	FCURAME	16	H'FFFFA854	8, 16
	IFS control mode register	IFSCMR	16	H'FFFFA890	8, 16
	Flash status register 0	FSTATR0	8	H'FFFFA900	8, 16
	Flash status register 1	FSTATR1	8	H'FFFFA901	8, 16
	Flash P/E mode entry register	FENTRYR	16	H'FFFFA902	8, 16
	Flash protect register	FPROTR	16	H'FFFFA904	8, 16
	Flash reset register	FRESETR	16	H'FFFFA906	8, 16
	FCU command register	FCMDR	16	H'FFFFA90A	8, 16
	FCU RAM ECC error control register	FRAMECCR	8	H'FFFFA90C	8
	IFS status register	IFSSR	8	H'FFFFA912	8
	FCU processing switch register	FCPSR	16	H'FFFFA918	8, 16
	EEPROM blank check control register	EEPBCCNT	16	H'FFFFA91A	8, 16

Module	Register Name	Abbreviation	Number of Bits	Address	Access Size
ROM/ IFS/ EEPROM	Flash P/E status register	FPESTAT	16	H'FFFFA91C	8, 16
	EEPROM blank check status register	EEPBCSTAT	16	H'FFFFA91E	8, 16
	EEPROM MAT select register	EEPMAT	16	H'FFFFAB00	8, 16
ROMC	ROM cache control register	RCCR	32	H'FFFC1400	32
	ROM cache control register 2	RCCR2	32	H'FFFC1408	32
RAM	RAM enable control register 0	RAMEN0	16	H'FFFF0800	8, (16)
	RAM write enable control register 0	RAMWEN0	16	H'FFFF0802	8, (16)
	RAM ECC enable control register	RAMECC	16	H'FFFF0804	8, (16)
	RAM error status register	RAMERR	8	H'FFFF0806	8
	RAM error interrupt control register	RAMINT	8	H'FFFF0810	8
	RAM access cycle set register	RAMACYC	16	H'FFFF0812	8, 16
	RAM enable control register 1	RAMEN1	16	H'FFFF0840	8, (16)
	RAM write enable control register 1	RAMWEN1	16	H'FFFF0842	8, (16)
Power-down mode	Standby control register	STBCR	16	H'FFFE0400	8, 16

33.2 List of Register Bits

The addresses and bit names of the registers in the peripheral modules are shown below. Registers composed of 16 and 32 bits are divided into 2 and 4 rows, accordingly.

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
INTC	ICR0	NMIL	—	—	—	—	—	—	NMIE
		—	—	—	—	—	—	—	—
	ICR1	IRQ71S	IRQ70S	IRQ61S	IRQ60S	IRQ51S	IRQ50S	IRQ41S	IRQ40S
		IRQ31S	IRQ30S	IRQ21S	IRQ20S	IRQ11S	IRQ10S	IRQ01S	IRQ00S
	IRQRR	—	—	—	—	—	—	—	—
		IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F
	IBCR	E15	E14	E13	E12	E11	E10	E9	E8
		E7	E6	E5	E4	E3	E2	E1	—
	IBNR	BE[1:0]		BOVE	—	—	—	—	—
		—	—	—	—	BN[3:0]			
	SINTR1	—	—	—	—	—	—	—	SINTC
	SINTR2	—	—	—	—	—	—	—	SINTC
	SINTR3	—	—	—	—	—	—	—	SINTC
	SINTR4	—	—	—	—	—	—	—	SINTC
	SINTR5	—	—	—	—	—	—	—	SINTC
	SINTR6	—	—	—	—	—	—	—	SINTC
	SINTR7	—	—	—	—	—	—	—	SINTC
	SINTR8	—	—	—	—	—	—	—	SINTC
	IPR01	IRQ0				IRQ1			
		IRQ2				IRQ3			
	IPR02	IRQ4				IRQ5			
		IRQ6				IRQ7			
	SINTR9	—	—	—	—	—	—	—	SINTC
	SINTR10	—	—	—	—	—	—	—	SINTC
	SINTR11	—	—	—	—	—	—	—	SINTC
	SINTR12	—	—	—	—	—	—	—	SINTC
	SINTR13	—	—	—	—	—	—	—	SINTC

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
INTC	SINTR14	—	—	—	—	—	—	—	SINTC
	SINTR15	—	—	—	—	—	—	—	SINTC
	IPR03	DMAC0				DMAC1			
		DMAC2				DMAC3			
	IPR04	DMAC4				DMAC5			
		DMAC6				DMAC7			
	IPR05	CMT0				CMT1			
		Reserved				WDT			
	IPR06	ATU-A (ICIA0, ICIA1)				ATU-A (ICIA2, ICIA3)			
		ATU-A (ICIA4, ICIA5)				ATU-A (OVIA)			
	IPR07	ATU-B (CMIB0, CMIB1)				ATU-B (CMIB6, ICIB0)			
		ATU-C0 (IMIC00 to IMIC03)				ATU-C0 (OVIC0)			
	IPR08	ATU-C1 (IMIC10 to IMIC13)				ATU-C1 (OVIC1)			
		ATU-C2 (IMIC20 to IMIC23)				ATU-C2 (OVIC2)			
	IPR09	ATU-C3 (IMIC30 to IMIC33)				ATU-C3 (OVIC3)			
		ATU-C4 (IMIC40 to IMIC43)				ATU-C4 (OVIC4)			
	IPR10	ATU-D0 (CMID00 to CMID03)				ATU-D0 (OVI1D0, OVI2D0)			
		ATU-D0 (UDID00 to UDID03)				ATU-D1 (CMID10 to CMID13)			
	IPR11	ATU-D1 (OVI1D1, OVI2D1)				ATU-D1 (UDID10 to UDID13)			
		ATU-D2 (CMID20 to CMID23)				ATU-D2 (OVI1D2, OVI2D2)			
	IPR12	ATU-D2 (UDID20 to UDID23)				ATU-D3 (CMID30 to CMID33)			
		ATU-D3 (OVI1D3, OVI2D3)				ATU-D3 (UDID30 to UDID33)			
	IPR13	Reserved				Reserved			
		Reserved				Reserved			
	IPR14	Reserved				Reserved			
		ATU-E0 (CMIE00 to CMIE03)				ATU-E1 (CMIE10 to CMIE13)			
	IPR15	ATU-E2 (CMIE20 to CMIE23)				ATU-E3 (CMIE30 to CMIE33)			
		ATU-E4 (CMIE40 to CMIE43)				ATU-E5 (CMIE50 to CMIE53)			
	IPR16	ATU-F (ICIF0 to ICIF3)				ATU-F (ICIF4 to ICIF7)			
		ATU-F (ICIF8 to ICIF11)				ATU-F (ICIF12 to ICIF15)			
	IPR17	ATU-F (ICIF16 to ICIF19)				ATU-F (ICIF20 to ICIF23)			
		ATU-F (ICIF24 to ICIF27)				Reserved			

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
INTC	IPR18	ATU-F (OVIF0 to OVIF3)				ATU-F (OVIF4 to OVIF7)			
		ATU-F (OVIF8 to OVIF11)				ATU-F (OVIF12 to OVIF15)			
	IPR19	ATU-F (OVIF16 to OVIF19)				ATU-F (OVIF20 to OVIF23)			
		ATU-F (OVIF24 to OVIF27)				Reserved			
	IPR20	ATU-G (CMIG0 to CMIG3)				ATU-G (CMIG4, CMIG5)			
		ATU-H (CMIH)				Reserved			
	IPR21	ATU-J (DFIJ0, DFIJ1)				ATU-J (OVIJ0, OVIJ1)			
		ATU-J (DOVIJ0, DOVIJ1)				ATU-E6 (CMIE60 to CMIE63)			
	IPR22	ADC (ADI0)				ADC (ADI1)			
		ADC (ADID0 to ADID3)				ADC (ADID4 to ADID7)			
	IPR23	ADC (ADID8 to ADID11)				ADC (ADID12 to ADID15)			
		ADC (ADID40)				ADC (ADID41)			
	IPR24	ADC (ADID42)				ADC (ADID43)			
		ADC (ADID44)				ADC (ADID45)			
	IPR25	ADC (ADID46)				ADC (ADID47)			
		Reserved				Reserved			
	IPR26	SCI_A				SCI_B			
		SCI_C				SCI_D			
	IPR27	SCI_E				RSPI_A			
		RSPI_B				RSPI_C			
	IPR28	RCAN_A				RCAN_B			
		RCAN_C				RCAN_D			
	IPR29	A-DMAC (TE74)				A-DMAC (TE75)			
		Reserved				Reserved			
	IPR30	FlexRay FR0				FlexRay FR1			
		FlexRay FRT0				FlexRay FRT1			
UBC	BAR0	BA0_31	BA0_30	BA0_29	BA0_28	BA0_27	BA0_26	BA0_25	BA0_24
		BA0_23	BA0_22	BA0_21	BA0_20	BA0_19	BA0_18	BA0_17	BA0_16
		BA0_15	BA0_14	BA0_13	BA0_12	BA0_11	BA0_10	BA0_9	BA0_8
		BA0_7	BA0_6	BA0_5	BA0_4	BA0_3	BA0_2	BA0_1	BA0_0

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
UBC	BAMR0	BAM0_31	BAM0_30	BAM0_29	BAM0_28	BAM0_27	BAM0_26	BAM0_25	BAM0_24
		BAM0_23	BAM0_22	BAM0_21	BAM0_20	BAM0_19	BAM0_18	BAM0_17	BAM0_16
		BAM0_15	BAM0_14	BAM0_13	BAM0_12	BAM0_11	BAM0_10	BAM0_9	BAM0_8
		BAM0_7	BAM0_6	BAM0_5	BAM0_4	BAM0_3	BAM0_2	BAM0_1	BAM0_0
	BBR0	—	—	UBID0	—	—	CP0_[2:0]		
		CD0_[1:0]		ID0_[1:0]		RW0_[1:0]		SZ0_[1:0]	
	BAR1	BA1_31	BA1_30	BA1_29	BA1_28	BA1_27	BA1_26	BA1_25	BA1_24
		BA1_23	BA1_22	BA1_21	BA1_20	BA1_19	BA1_18	BA1_17	BA1_16
		BA1_15	BA1_14	BA1_13	BA1_12	BA1_11	BA1_10	BA1_9	BA1_8
		BA1_7	BA1_6	BA1_5	BA1_4	BA1_3	BA1_2	BA1_1	BA1_0
	BAMR1	BAM1_31	BAM1_30	BAM1_29	BAM1_28	BAM1_27	BAM1_26	BAM1_25	BAM1_24
		BAM1_23	BAM1_22	BAM1_21	BAM1_20	BAM1_19	BAM1_18	BAM1_17	BAM1_16
		BAM1_15	BAM1_14	BAM1_13	BAM1_12	BAM1_11	BAM1_10	BAM1_9	BAM1_8
		BAM1_7	BAM1_6	BAM1_5	BAM1_4	BAM1_3	BAM1_2	BAM1_1	BAM1_0
	BBR1	—	—	UBID1	—	—	CP1_[2:0]		
		CD1_[1:0]		ID1_[1:0]		RW1_[1:0]		SZ1_[1:0]	
	BAR2	BA2_31	BA2_30	BA2_29	BA2_28	BA2_27	BA2_26	BA2_25	BA2_24
		BA2_23	BA2_22	BA2_21	BA2_20	BA2_19	BA2_18	BA2_17	BA2_16
		BA2_15	BA2_14	BA2_13	BA2_12	BA2_11	BA2_10	BA2_9	BA2_8
		BA2_7	BA2_6	BA2_5	BA2_4	BA2_3	BA2_2	BA2_1	BA2_0
	BAMR2	BAM2_31	BAM2_30	BAM2_29	BAM2_28	BAM2_27	BAM2_26	BAM2_25	BAM2_24
		BAM2_23	BAM2_22	BAM2_21	BAM2_20	BAM2_19	BAM2_18	BAM2_17	BAM2_16
		BAM2_15	BAM2_14	BAM2_13	BAM2_12	BAM2_11	BAM2_10	BAM2_9	BAM2_8
		BAM2_7	BAM2_6	BAM2_5	BAM2_4	BAM2_3	BAM2_2	BAM2_1	BAM2_0
	BBR2	—	—	UBID2	—	—	CP2_[2:0]		
		CD2_[1:0]		ID2_[1:0]		RW2_[1:0]		SZ2_[1:0]	
	BAR3	BA3_31	BA3_30	BA3_29	BA3_28	BA3_27	BA3_26	BA3_25	BA3_24
		BA3_23	BA3_22	BA3_21	BA3_20	BA3_19	BA3_18	BA3_17	BA3_16
		BA3_15	BA3_14	BA3_13	BA3_12	BA3_11	BA3_10	BA3_9	BA3_8
		BA3_7	BA3_6	BA3_5	BA3_4	BA3_3	BA3_2	BA3_1	BA3_0

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
UBC	BAMR3	BAM3_31	BAM3_30	BAM3_29	BAM3_28	BAM3_27	BAM3_26	BAM3_25	BAM3_24
		BAM3_23	BAM3_22	BAM3_21	BAM3_20	BAM3_19	BAM3_18	BAM3_17	BAM3_16
		BAM3_15	BAM3_14	BAM3_13	BAM3_12	BAM3_11	BAM3_10	BAM3_9	BAM3_8
		BAM3_7	BAM3_6	BAM3_5	BAM3_4	BAM3_3	BAM3_2	BAM3_1	BAM3_0
	BBR3	—	—	UBID3	—	—	CP3_[2:0]		
		CD3_[1:0]		ID3_[1:0]		RW3_[1:0]		SZ3_[1:0]	
	BRCR	—	—	—	—	—	—	—	—
		—	—	UTOD3	UTOD2	UTOD1	UTOD0	CKS[1:0]	
		SCMFC0	SCMFC1	SCMFC2	SCMFC3	SCMFD0	SCMFD1	SCMFD2	SCMFD3
		PCB3	PCB2	PCB1	PCB0	—	—	—	—
	BAR4	BA4_31	BA4_30	BA4_29	BA4_28	BA4_27	BA4_26	BA4_25	BA4_24
		BA4_23	BA4_22	BA4_21	BA4_20	BA4_19	BA4_18	BA4_17	BA4_16
		BA4_15	BA4_14	BA4_13	BA4_12	BA4_11	BA4_10	BA4_9	BA4_8
		BA4_7	BA4_6	BA4_5	BA4_4	BA4_3	BA4_2	BA4_1	BA4_0
	BAMR4	BAM4_31	BAM4_30	BAM4_29	BAM4_28	BAM4_27	BAM4_26	BAM4_25	BAM4_24
		BAM4_23	BAM4_22	BAM4_21	BAM4_20	BAM4_19	BAM4_18	BAM4_17	BAM4_16
		BAM4_15	BAM4_14	BAM4_13	BAM4_12	BAM4_11	BAM4_10	BAM4_9	BAM4_8
		BAM4_7	BAM4_6	BAM4_5	BAM4_4	BAM4_3	BAM4_2	BAM4_1	BAM4_0
	BBR4	—	—	UBID4	—	—	CP4_[2:0]		
		CD4_[1:0]		ID4_[1:0]		RW4_[1:0]		SZ4_[1:0]	
	BAR5	BA5_31	BA5_30	BA5_29	BA5_28	BA5_27	BA5_26	BA5_25	BA5_24
		BA5_23	BA5_22	BA5_21	BA5_20	BA5_19	BA5_18	BA5_17	BA5_16
		BA5_15	BA5_14	BA5_13	BA5_12	BA5_11	BA5_10	BA5_9	BA5_8
		BA5_7	BA5_6	BA5_5	BA5_4	BA5_3	BA5_2	BA5_1	BA5_0
	BAMR5	BAM5_31	BAM5_30	BAM5_29	BAM5_28	BAM5_27	BAM5_26	BAM5_25	BAM5_24
		BAM5_23	BAM5_22	BAM5_21	BAM5_20	BAM5_19	BAM5_18	BAM5_17	BAM5_16
		BAM5_15	BAM5_14	BAM5_13	BAM5_12	BAM5_11	BAM5_10	BAM5_9	BAM5_8
		BAM5_7	BAM5_6	BAM5_5	BAM5_4	BAM5_3	BAM5_2	BAM5_1	BAM5_0
	BBR5	—	—	UBID5	—	—	CP5_[2:0]		
		CD5_[1:0]		ID5_[1:0]		RW5_[1:0]		SZ5_[1:0]	

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
UBC	BAR6	BA6_31	BA6_30	BA6_29	BA6_28	BA6_27	BA6_26	BA6_25	BA6_24
		BA6_23	BA6_22	BA6_21	BA6_20	BA6_19	BA6_18	BA6_17	BA6_16
		BA6_15	BA6_14	BA6_13	BA6_12	BA6_11	BA6_10	BA6_9	BA6_8
		BA6_7	BA6_6	BA6_5	BA6_4	BA6_3	BA6_2	BA6_1	BA6_0
	BAMR6	BAM6_31	BAM6_30	BAM6_29	BAM6_28	BAM6_27	BAM6_26	BAM6_25	BAM6_24
		BAM6_23	BAM6_22	BAM6_21	BAM6_20	BAM6_19	BAM6_18	BAM6_17	BAM6_16
		BAM6_15	BAM6_14	BAM6_13	BAM6_12	BAM6_11	BAM6_10	BAM6_9	BAM6_8
		BAM6_7	BAM6_6	BAM6_5	BAM6_4	BAM6_3	BAM6_2	BAM6_1	BAM6_0
	BBR6	—	—	UBID6	—	—	CP6_[2:0]		
		CD6_[1:0]		ID6_[1:0]		RW6_[1:0]		SZ6_[1:0]	
	BAR7	BA7_31	BA7_30	BA7_29	BA7_28	BA7_27	BA7_26	BA7_25	BA7_24
		BA7_23	BA7_22	BA7_21	BA7_20	BA7_19	BA7_18	BA7_17	BA7_16
		BA7_15	BA7_14	BA7_13	BA7_12	BA7_11	BA7_10	BA7_9	BA7_8
		BA7_7	BA7_6	BA7_5	BA7_4	BA7_3	BA7_2	BA7_1	BA7_0
	BAMR7	BAM7_31	BAM7_30	BAM7_29	BAM7_28	BAM7_27	BAM7_26	BAM7_25	BAM7_24
		BAM7_23	BAM7_22	BAM7_21	BAM7_20	BAM7_19	BAM7_18	BAM7_17	BAM7_16
		BAM7_15	BAM7_14	BAM7_13	BAM7_12	BAM7_11	BAM7_10	BAM7_9	BAM7_8
		BAM7_7	BAM7_6	BAM7_5	BAM7_4	BAM7_3	BAM7_2	BAM7_1	BAM7_0
	BBR7	—	—	UBID7	—	—	CP7_[2:0]		
		CD7_[1:0]		ID7_[1:0]		RW7_[1:0]		SZ7_[1:0]	
	BRCR1	—	—	—	—	—	—	—	—
		—	—	UTOD7	UTOD6	UTOD5	UTOD4	—	—
		SCMFC4	SCMFC5	SCMFC6	SCMFC7	SCMFD4	SCMFD5	SCMFD6	SCMFD7
		PCB7	PCB6	PCB5	PCB4	—	—	—	—
BSC	CS0BCR	—	IWW[2:0]			IWRWD[2:0]			IWRWS[2]
		IWRWS[1:0]		IWRRD[2:0]			IWRRS[2:0]		
		—	—	TYPE[1:0]		—	BSZ[1:0]		—
		—	—	—	—	—	—	—	—
	CS1BCR	—	IWW[2:0]			IWRWD[2:0]			IWRWS[2]
		IWRWS[1:0]		IWRRD[2:0]			IWRRS[2:0]		
		—	—	TYPE[1:0]		—	BSZ[1:0]		—
		—	—	—	—	—	—	—	—

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
BSC	CS2BCR	—	IWW[2:0]			IWRWD[2:0]			IWRWS[2]
		IWRWS[1:0]		IWRRD[2:0]			IWRRS[2:0]		
		—	—	TYPE[1:0]		—	BSZ[1:0]		—
		—	—	—	—	—	—	—	—
	CS3BCR	—	IWW[2:0]			IWRWD[2:0]			IWRWS[2]
		IWRWS[1:0]		IWRRD[2:0]			IWRRS[2:0]		
		—	—	TYPE[1:0]		—	BSZ[1:0]		—
		—	—	—	—	—	—	—	—
	CS0WCR	—	—	—	—	—	—	—	—
		—	—	—	BAS	—	WW[2:0]		
		—	—	—	SW[1:0]		WR[3:1]		
		WR[0]	WM	—	—	—	—	HW[1:0]	
	CS1WCR	—	—	—	—	—	—	—	—
		—	—	—	BAS	—	WW[2:0]		
		—	—	—	SW[1:0]		WR[3:1]		
		WR[0]	WM	—	—	—	—	HW[1:0]	
	CS2WCR	—	—	—	—	—	—	—	—
		—	—	—	BAS	—	WW[2:0]		
		—	—	—	SW[1:0]		WR[3:1]		
		WR[0]	WM	—	—	—	—	HW[1:0]	
	CS3WCR	—	—	—	—	—	—	—	—
		—	—	—	BAS	—	WW[2:0]		
		—	—	—	SW[1:0]		WR[3:1]		
		WR[0]	WM	—	—	—	—	HW[1:0]	
DMAC	SAR0								
	DAR0								

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
DMAC	DMATCR0	—	—	—	—	—	—	—	—
	CHCR0	TC[1:0]		—	RLD1	RLD2[1:0]		—	IFT
		—	—	—	—	—	HIE	—	—
		DM[1:0]		SM[1:0]		RS[3:0]			
		—	—	TB	TS[1:0]		IE	—	DE
	CHFR0	—	—	—	HE	—	—	—	TE
	TEMSK0	TEMKEY[7:0]							
		—	—	—	—	—	—	—	TEMASK
	RSAR0								
	RDAR0								
	RDMATCR0	—	—	—	—	—	—	—	—
	SAR1								
	DAR1								

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
DMAC	DMATCR1	—	—	—	—	—	—	—	—
	CHCR1	TC[1:0]		—	RLD1	RLD2[1:0]		—	IFT
		—	—	—	—	—	HIE	—	—
		DM[1:0]		SM[1:0]		RS[3:0]			
		—	—	TB	TS[1:0]		IE	—	DE
	CHFR1	—	—	—	HE	—	—	—	TE
	TEMSK1	TEMKEY[7:0]							
		—	—	—	—	—	—	—	TEMASK
	RSAR1								
	RDAR1								
	RDMATCR1	—	—	—	—	—	—	—	—
	SAR2								
	DAR2								

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
DMAC	DMATCR2	—	—	—	—	—	—	—	—
	CHCR2	TC[1:0]		—	RLD1	RLD2[1:0]		—	IFT
		—	—	—	—	—	HIE	—	—
		DM[1:0]		SM[1:0]		RS[3:0]			
		—	—	TB	TS[1:0]		IE	—	DE
	CHFR2	—	—	—	HE	—	—	—	TE
	TEMSK2	TEMKEY[7:0]							
		—	—	—	—	—	—	—	TEMASK
	RSAR2								
	RDAR2								
	RDMATCR2	—	—	—	—	—	—	—	—
	SAR3								
	DAR3								

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
DMAC	DMATCR3	—	—	—	—	—	—	—	—
	CHCR3	TC[1:0]		—	RLD1	RLD2[1:0]		—	IFT
		—	—	—	—	—	HIE	—	—
		DM[1:0]		SM[1:0]		RS[3:0]			
		—	—	TB	TS[1:0]		IE	—	DE
	CHFR3	—	—	—	HE	—	—	—	TE
	TEMSK3	TEMKEY[7:0]							
		—	—	—	—	—	—	—	TEMASK
	RSAR3								
	RDAR3								
	RDMATCR3	—	—	—	—	—	—	—	—
	SAR4								
	DAR4								

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
DMAC	DMATCR4	—	—	—	—	—	—	—	—
	CHCR4	TC[1:0]		—	RLD1	RLD2[1:0]		—	IFT
		—	—	—	—	—	HIE	—	—
		DM[1:0]		SM[1:0]		RS[3:0]			
		—	—	TB	TS[1:0]		IE	—	DE
	CHFR4	—	—	—	HE	—	—	—	TE
	TEMSK4	TEMKEY[7:0]							
		—	—	—	—	—	—	—	TEMASK
	RSAR4								
	RDAR4								
	RDMATCR4	—	—	—	—	—	—	—	—
	ARCR4	—	—	—	—	—	—	—	—
	RARCR4	—	—	—	—	—	—	—	—
	SAR5								

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
DMAC	DAR5								
	DMATCR5	—	—	—	—	—	—	—	—
	CHCR5	TC[1:0]		—	RLD1	RLD2[1:0]		—	IFT
		—	—	—	—	—	HIE	—	—
		DM[1:0]		SM[1:0]		RS[3:0]			
		—	—	TB	TS[1:0]		IE	—	DE
	CHFR5	—	—	—	HE	—	—	—	TE
	TEMSK5	TEMKEY[7:0]							
		—	—	—	—	—	—	—	TEMASK
	RSAR5								
	RDAR5								
	RDMATCR5	—	—	—	—	—	—	—	—
	ARCR5	—	—	—	—	—	—	—	—
	RARCR5	—	—	—	—	—	—	—	—

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
DMAC	SAR6								
	DAR6								
	DMATCR6	—	—	—	—	—	—	—	—
	CHCR6	TC[1:0]		—	RLD1	RLD2[1:0]		—	IFT
		—	—	—	—	—	HIE	—	—
		DM[1:0]		SM[1:0]		RS[3:0]			
		—	—	TB	TS[1:0]		IE	—	DE
	CHFR6	—	—	—	HE	—	—	—	TE
	TEMSK6	TEMKEY[7:0]							
		—	—	—	—	—	—	—	TEMASK
	RSAR6								
	RDAR6								
	RDMATCR6	—	—	—	—	—	—	—	—

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
DMAC	ARCR6	—	—	—	—	—	—	—	—
	RARCR6	—	—	—	—	—	—	—	—
	SAR7								
	DAR7								
	DMATCR7	—	—	—	—	—	—	—	—
	CHCR7	TC[1:0]		—	RLD1	RLD2[1:0]		—	IFT
		—	—	—	—	—	HIE	—	—
		DM[1:0]		SM[1:0]		RS[3:0]			
		—	—	TB	TS[1:0]		IE	—	DE
	CHFR7	—	—	—	HE	—	—	—	TE
	TEMSK7	TEMKEY[7:0]							
		—	—	—	—	—	—	—	TEMASK
	RSAR7								
	RDAR7								

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
DMAC	RDMATCR7	—	—	—	—	—	—	—	—
	ARCR7	—	—	—	—	—	—	—	—
	RARCR7	—	—	—	—	—	—	—	—
	DMAOR	—	—	CMS[1:0]		—	—	PR[1:0]	
		—	—	—	—	—	—	—	DME
	DMAFR	—	—	—	AE	—	—	—	NMIF
	DMARS0	CH1 MID[5:0]						CH1 RID[1:0]	
		CH0 MID[5:0]						CH0 RID[1:0]	
	DMARS1	CH3 MID[5:0]						CH3 RID[1:0]	
		CH2 MID[5:0]						CH2 RID[1:0]	
	DMARS2	CH5 MID[5:0]						CH5 RID[1:0]	
		CH4 MID[5:0]						CH4 RID[1:0]	
	DMARS3	CH7 MID[5:0]						CH7 RID[1:0]	
		CH6 MID[5:0]						CH6 RID[1:0]	
A-DMAC	ADMAOR	—	—	—	—	—	—	—	DME
	ADMAABR	—	—	—	—	—	AA[2:0]		
	ADMAIE0							—	—
	ADMAIE1								
	ADMAIE2								
	ADMAIE3								
	ADMAIE4								
	ADMAIE5								
	ADMAIE6								
	ADMAIE7								
	ADMAIE8								
	ADMAIE9	—	—	—	—			—	—
	ADMADV0							—	—

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
A-DMAC	ADMADV1								
	ADMADV2								
	ADMADV3								
	ADMADV4								
	ADMADV5								
	ADMADV6								
	ADMATE0								
	ADMATE1								
	ADMATE2	—	—	—	—			—	—
	ADMADE0							—	—
	ADMADE1								
	ADMADE2								
	ADMADE3								
	ADMADE4								
	ADMADE5								
	ADMADE6								
	ADMADE7	—	—						
	ADMAMODE0					—	—	—	—
	ADMAMODE1								
	ADMAMODE2								
	ADMAMODE3								
	ADMATCR0	—	—	—	—	—	—		
	ADMARTCR0	—	—	—	—	—	—		
	ADMATCR1	—	—	—	—	—	—		
	ADMARTCR1	—	—	—	—	—	—		
	ADMATCR56	—	—	—	—	—	—		

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
A-DMAC	ADMATCR57	—	—	—	—	—	—		
	ADMATCR58	—	—	—	—	—	—		
	ADMATCR59	—	—	—	—	—	—		
	ADMATCR60	—	—	—	—	—	—		
	ADMATCR61	—	—	—	—	—	—		
	ADMATCR62	—	—	—	—	—	—		
	ADMATCR63	—	—	—	—	—	—		
	ADMATCR64	—	—	—	—	—	—		
	ADMATCR65	—	—	—	—	—	—		
	ADMATCR66	—	—	—	—	—	—		
	ADMATCR67	—	—	—	—	—	—		
	ADMATCR68	—	—	—	—	—	—		
	ADMATCR69	—	—	—	—	—	—		
	ADMATCR70	—	—	—	—	—	—		
	ADMATCR71	—	—	—	—	—	—		
	ADMAAR0	—							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
A-DMAC	ADMARAR0	—							
	ADMAAR1	—							
	ADMARAR1	—							
	ADMAAR56	—							
	ADMAAR57	—							
	ADMAAR58	—							
	ADMAAR59	—							
	ADMAAR60	—							
	ADMAAR61	—							
	ADMAAR62	—							
	ADMAAR63	—							
	ADMAAR64	—							
	ADMAAR65	—							
	ADMAAR66	—							
	ADMAAR67	—							
	ADMAAR68	—							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
A-DMAC	ADMAAR69	—							
	ADMAAR70	—							
	ADMAAR71	—							
	ADMABUF2								
	ADMABUF3								
	ADMABUF4								
	ADMABUF5								
	ADMABUF6								
	ADMABUF7								

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
A-DMAC	ADMARVPR0								
	ADMARVPR1								
	ADMARVPR2								
	ADMARVPR3								
	ADMARVPR4								
	ADMARVPR5								
	ADMARVPR6								
	ADMARVPR7								
	ADMATVPR0								
	ADMATVPR1								
	ADMATVPR2								
	ADMATVPR3								
	ADMATVPR4								
	ADMATVPR5								
	ADMATVPR6								
	ADMATVPR7								

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
A-DMAC	ADMAFRWR0								
	ADMAFRWR1								
	ADMAFRWR2								
	ADMAFRWR3								
	ADMAFRLMB	—	LADMB6	LADMB5	LADMB4	LADMB3	LADMB2	LADMB1	LADMB0
	ADMAFRGENCTR	—	—	—	—	—	—	—	ADPEHM
	ADMAFRCTR	ADLNG[1:0]		—	—	—	—	—	ADENC
	ADMAFRFCTR	ADFLNG[1:0]		—	—	—	—	—	ADFENC
	ADMAFRTRSTAT	ADTMB7	ADTMB6	ADTMB5	ADTMB4	ADTMB3	ADTMB2	ADTMB1	ADTMB0
	ADMAFRSTAT	—	—	—	—	—	—	ADDPE	ADSTAT
	ADMAFRFSTAT	—	—	—	—	—	FRWF	ADFDPE	ADFSTAT
ATU-III	ATUENR	—	—	—	—	—	—	TJE	THE
		TGE	TFE	TEE	TDE	TCE	TBE	TAE	PSCE
	CBCNT	—	—	CB4EG[1:0]		—	CB5SEL	CB5EG[1:0]	
	NCMR	NCCSEL	—	—	—	NCMJ	NCMF	NCMC	NCMA
	PSCR0	—	—	—	—	—	—	PSC0[9:8]	
		PSC0[7:0]							
	PSCR1	—	—	—	—	—	—	PSC1[9:8]	
		PSC1[7:0]							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	PSCR2	—	—	—	—	—	—	PSC2[9:8]	
		PSC2[7:0]							
	PSCR3	—	—	—	—	—	—	PSC3[9:8]	
		PSC3[7:0]							
	TCRA	EVOSEL2A	EVOSEL2B	EVOSEL1			CKSELA		
	TIOR1A	—	—	—	—	IOA5		IOA4	
		IOA3		IOA2		IOA1		IOA0	
	TIOR2A	—	—	NCKA5	NCKA4	NCKA3	NCKA2	NCKA1	NCKA0
		—	—	NCEA5	NCEA4	NCEA3	NCEA2	NCEA1	NCEA0
	TSRA	OVFA	—	ICFA5	ICFA4	ICFA3	ICFA2	ICFA1	ICFA0
	TIERA	OVEA	—	ICEA5	ICEA4	ICEA3	ICEA2	ICEA1	ICEA0
	NCMCR1A	—	—	NCM1A5	NCM1A4	NCM1A3	NCM1A2	NCM1A1	NCM1A0
	NCNTA0	NCCNTA0[7:0]							
	NCRA0	NCTA0[7:0]							
	NCNTA1	NCCNTA1[7:0]							
	NCRA1	NCTA1[7:0]							
	NCNTA2	NCCNTA2[7:0]							
	NCRA2	NCTA2[7:0]							
	NCNTA3	NCCNTA3[7:0]							
	NCRA3	NCTA3[7:0]							
	NCNTA4	NCCNTA4[7:0]							
	NCRA4	NCTA4[7:0]							
	NCNTA5	NCCNTA5[7:0]							
	NCRA5	NCTA5[7:0]							
	TCNTA	CNTA[31:24]							
		CNTA[23:16]							
		CNTA[15:8]							
		CNTA[7:0]							
	ICRA0	ICA0[31:24]							
		ICA0[23:16]							
		ICA0[15:8]							
		ICA0[7:0]							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	ICRA1	ICA1[31:24]							
		ICA1[23:16]							
		ICA1[15:8]							
		ICA1[7:0]							
	ICRA2	ICA2[31:24]							
		ICA2[23:16]							
		ICA2[15:8]							
		ICA2[7:0]							
	ICRA3	ICA3[31:24]							
		ICA3[23:16]							
		ICA3[15:8]							
		ICA3[7:0]							
	ICRA4	ICA4[31:24]							
		ICA4[23:16]							
		ICA4[15:8]							
		ICA4[7:0]							
	ICRA5	ICA5[31:24]							
		ICA5[23:16]							
		ICA5[15:8]							
		ICA5[7:0]							
	TCRB	—	—	—	—	—	—	CKSELB	
	TIORB	LDSEL	CTCNTB5	EVCNTB	LDEN	CCS	—	—	IOB6
	TSRB	—	—	—	—	CMFB6	CMFB1	ICFB0	CMFB0
	TIERB	—	—	IREG		CMEB6	CMEB1	ICEB0	CMEB0
	TCNTB0	CNTB0							
		CNTB0							
		CNTB0							
		CNTB0							
	ICRB0	ICB0							
		ICB0							
		ICB0							
		ICB0							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	OCRB0	OCB0							
		OCB0							
		OCB0							
		OCB0							
	TCNTB1	CNTB1							
	OCRB1	OCB1							
	ICRB1	ICB1							
		ICB1							
		ICB1							
		ICB1							
	ICRB2	ICB2							
		ICB2							
		ICB2							
		ICB2							
	LDB	—	—	—	—	—	—	—	—
		LDVAL							
		LDVAL							
		LDVAL							
	RLDB	RLDVAL							
		RLDVAL							
		RLDVAL							
		—	—	—	—	—	—	—	—
	PIMR	—	—	—	—	PIM			
		PIM							
	TCNTB2	CNTB2							
		CNTB2							
		CNTB2							
		—	—	—	—	—	—	—	—
	TCNTB6	CNTB6							
		CNTB6							
		CNTB6				—	—	—	—
		—	—	—	—	—	—	—	—

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	OCRB6	OCB6							
		OCB6							
		OCB6				—	—	—	—
		—	—	—	—	—	—	—	—
	OCRB7	OCB7							
		OCB7							
		OCB7				—	—	—	—
		—	—	—	—	—	—	—	—
	TCNTB3	CNTB3							
		CNTB3							
		CNTB3				—	—	—	—
		—	—	—	—	—	—	—	—
	TCNTB4	CNTB4							
		CNTB4							
		CNTB4				—	—	—	—
		—	—	—	—	—	—	—	—
	TCNTB5	CNTB5							
		CNTB5							
		CNTB5				—	—	—	—
		—	—	—	—	—	—	—	—
	TCCLRB	CCLRB							
		CCLRB							
		CCLRB				—	—	—	—
		—	—	—	—	—	—	—	—
	TSTRC	—	—	—	STRC4	STRC3	STRC2	STRC1	STRC0
	NCCRC0	—	—	—	—	NCEC03	NCEC02	NCEC01	NCEC00
	NCCRC1	—	—	—	—	NCEC13	NCEC12	NCEC11	NCEC10
	NCCRC2	—	—	—	—	NCEC23	NCEC22	NCEC21	NCEC20
	NCCRC3	—	—	—	—	NCEC33	NCEC32	NCEC31	NCEC30
	NCCRC4	—	—	—	—	NCEC43	NCEC42	NCEC41	NCEC40
	NCNTC00	NCNTC00[7:0]							
	NCNTC01	NCNTC01[7:0]							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	NCNTC02	NCNTC02[7:0]							
	NCNTC03	NCNTC03[7:0]							
	NCRC00	NCRC00[7:0]							
	NCRC01	NCRC01[7:0]							
	NCRC02	NCRC02[7:0]							
	NCRC03	NCRC03[7:0]							
	NCNTC10	NCNTC10[7:0]							
	NCNTC11	NCNTC11[7:0]							
	NCNTC12	NCNTC12[7:0]							
	NCNTC13	NCNTC13[7:0]							
	NCRC10	NCRC10[7:0]							
	NCRC11	NCRC11[7:0]							
	NCRC12	NCRC12[7:0]							
	NCRC13	NCRC13[7:0]							
	NCNTC20	NCNTC20[7:0]							
	NCNTC21	NCNTC21[7:0]							
	NCNTC22	NCNTC22[7:0]							
	NCNTC23	NCNTC23[7:0]							
	NCRC20	NCRC20[7:0]							
	NCRC21	NCRC21[7:0]							
	NCRC22	NCRC22[7:0]							
	NCRC23	NCRC23[7:0]							
	NCNTC30	NCNTC30[7:0]							
	NCNTC31	NCNTC31[7:0]							
	NCNTC32	NCNTC32[7:0]							
	NCNTC33	NCNTC33[7:0]							
	NCRC30	NCRC30[7:0]							
	NCRC31	NCRC31[7:0]							
	NCRC32	NCRC32[7:0]							
	NCRC33	NCRC33[7:0]							
	NCNTC40	NCNTC40[7:0]							
	NCNTC41	NCNTC41[7:0]							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	NCNTC42	NCNTC42[7:0]							
	NCNTC43	NCNTC43[7:0]							
	NCRC40	NCRC40[7:0]							
	NCRC41	NCRC41[7:0]							
	NCRC42	NCRC42[7:0]							
	NCRC43	NCRC43[7:0]							
	TCRC0	FCMC03	FCMC02	FCMC01	FCMC00	PWM00	CKSELC0[2:0]		
	TIERC0	—	—	—	OVEC0	IMEC03	IMEC02	IMEC01	IMEC00
	TIORC0	—	IOC03[2:0]			—	IOC02[2:0]		
		—	IOC01[2:0]			—	IOC00[2:0]		
	TSRC0	—	—	—	OVFC0	IMFC03	IMFC02	IMFC01	IMFC00
	GRC00								
		—	—	—	—	—	—	—	—
	GRC01								
		—	—	—	—	—	—	—	—
	GRC02								
		—	—	—	—	—	—	—	—
	GRC03								
		—	—	—	—	—	—	—	—
	TCNTC0								
		—	—	—	—	—	—	—	—

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	TCRC1	FCMC13	FCMC12	FCMC11	FCMC10	PWM10	CKSELC1[2:0]		
	TIERC1	—	—	—	OVEC1	IMEC13	IMEC12	IMEC11	IMEC10
	TIORC1	—	IOC13[2:0]			—	IOC12[2:0]		
		—	IOC11[2:0]			—	IOC10[2:0]		
	TSRC1	—	—	—	OVFC1	IMFC13	IMFC12	IMFC11	IMFC10
	GRC10								
		—	—	—	—	—	—	—	—
	GRC11								
		—	—	—	—	—	—	—	—
	GRC12								
		—	—	—	—	—	—	—	—
	GRC13								
		—	—	—	—	—	—	—	—
	TCNTC1								
		—	—	—	—	—	—	—	—
	TCRC2	FCMC23	FCMC22	FCMC21	FCMC20	PWM20	CKSELC2[2:0]		
	TIERC2	—	—	—	OVEC2	IMEC23	IMEC22	IMEC21	IMEC20
	TIORC2	—	IOC23[2:0]			—	IOC22[2:0]		
		—	IOC21[2:0]			—	IOC20[2:0]		
	TSRC2	—	—	—	OVFC2	IMFC23	IMFC22	IMFC21	IMFC20

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	GRC20								
		—	—	—	—	—	—	—	—
	GRC21								
		—	—	—	—	—	—	—	—
	GRC22								
		—	—	—	—	—	—	—	—
	GRC23								
		—	—	—	—	—	—	—	—
	TCNTC2								
		—	—	—	—	—	—	—	—
	TCRC3	FCMC33	FCMC32	FCMC31	FCMC30	PWM30	CKSELC3[2:0]		
	TIERC3	—	—	—	OVEC3	IMEC33	IMEC32	IMEC31	IMEC30
	TIORC3	—	IOC33[2:0]			—	IOC32[2:0]		
		—	IOC31[2:0]			—	IOC30[2:0]		
	TSRC3	—	—	—	OVFC3	IMFC33	IMFC32	IMFC31	IMFC30
	GRC30								
		—	—	—	—	—	—	—	—

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	GRC31								
		—	—	—	—	—	—	—	—
	GRC32								
		—	—	—	—	—	—	—	—
	GRC33								
		—	—	—	—	—	—	—	—
	TCNTC3								
		—	—	—	—	—	—	—	—
	TCRC4	FCMC43	FCMC42	FCMC41	FCMC40	PWM40	CKSELC4[2:0]		
	TIERC4	—	—	—	OVEC4	IMEC43	IMEC42	IMEC41	IMEC40
	TIORC4	—	IOC43[2:0]			—	IOC42[2:0]		
		—	IOC41[2:0]			—	IOC40[2:0]		
	TSRC4	—	—	—	OVFC4	IMFC43	IMFC42	IMFC41	IMFC40
	GRC40								
		—	—	—	—	—	—	—	—
	GRC41								
		—	—	—	—	—	—	—	—

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	GRC42								
		—	—	—	—	—	—	—	—
	GRC43								
		—	—	—	—	—	—	—	—
	TCNTC4								
		—	—	—	—	—	—	—	—
	TSTRD	—	—	—	—	STRD3	STRD2	STRD1	STRD0
	TCNT1D0								
		—	—	—	—	—	—	—	—
	TCNT2D0								
		—	—	—	—	—	—	—	—
	OSBRD0								
		—	—	—	—	—	—	—	—
	TCRD0	—	OBRED0	C2CED0	C1CED0	—	CKSEL2D0[2:0]		
		—	CKSEL1D0[2:0]			—	DCSELD0[2:0]		
	TOCRD0	—	—	—	—	—	—	TONEBD0	TONEAD0
	CMPOD0	CMPBD03	CMPBD02	CMPBD01	CMPBD00	CMPAD03	CMPAD02	CMPAD01	CMPAD00

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	TCNT1D1								
		—	—	—	—	—	—	—	—
	TCNT2D1								
		—	—	—	—	—	—	—	—
	OSBRD1								
		—	—	—	—	—	—	—	—
	TCRD1	—	OBRED1	C2CED1	C1CED1	—	CKSEL2D1[2:0]		
		—	CKSEL1D1[2:0]			—	DCSELD1[2:0]		
	TOCRD1	—	—	—	—	—	—	TONEBD1	TONEAD1
	CMPOD1	CMPBD13	CMPBD12	CMPBD11	CMPBD10	CMPAD13	CMPAD12	CMPAD11	CMPAD10
	TCNT1D2								
		—	—	—	—	—	—	—	—
	TCNT2D2								
		—	—	—	—	—	—	—	—
	OSBRD2								
		—	—	—	—	—	—	—	—
	TCRD2	—	OBRED2	C2CED2	C1CED2	—	CKSEL2D2[2:0]		
		—	CKSEL1D2[2:0]			—	DCSELD2[2:0]		
	TOCRD2	—	—	—	—	—	—	TONEBD2	TONEAD2

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	TCNT1D3								
		—	—	—	—	—	—	—	—
	TCNT2D3								
		—	—	—	—	—	—	—	—
	OSBRD3								
		—	—	—	—	—	—	—	—
	TCRD3	—	OBRED3	C2CED3	C1CED3	—	CKSEL2D3[2:0]		
		—	CKSEL1D3[2:0]			—	DCSELD3[2:0]		
	TOCRD3	—	—	—	—	—	—	TONEBD3	TONEAD3
	TIOR1D0	OSSD03[1:0]		OSSD02[1:0]		OSSD01[1:0]		OSSD00[1:0]	
		IOAD03[1:0]		IOAD02[1:0]		IOAD01[1:0]		IOAD00[1:0]	
	TIOR2D0	—	IOBD03[2:0]			—	IOBD02[2:0]		
		—	IOBD01[2:0]			—	IOBD00[2:0]		
	DSTRD0	—	—	—	—	DSTD03	DSTD02	DSTD01	DSTD00
	DSRD0	—	—	—	—	DSFD03	DSFD02	DSFD01	DSFD00
	DCRD0	—	TRGSELD03[2:0]			—	TRGSELD02[2:0]		
		—	TRGSELD01[2:0]			—	TRGSELD00[2:0]		
	TSRD0	—	—	OVF2D0	OVF1D0	UDFD03	UDFD02	UDFD01	UDFD00
		CMFAD03	CMFAD02	CMFAD01	CMFAD00	CMFBD03	CMFBD02	CMFBD01	CMFBD00
	TIERD0	—	—	OVE2D0	OVE1D0	UDED03	UDED02	UDED01	UDED00
		CMEAD03	CMEAD02	CMEAD01	CMEAD00	CMEBD03	CMEBD02	CMEBD01	CMEBD00
	OCRD00								
		—	—	—	—	—	—	—	—

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	OCRD01								
		—	—	—	—	—	—	—	—
	OCRD02								
		—	—	—	—	—	—	—	—
	OCRD03								
		—	—	—	—	—	—	—	—
	GRD00								
		—	—	—	—	—	—	—	—
	GRD01								
		—	—	—	—	—	—	—	—
	GRD02								
		—	—	—	—	—	—	—	—
	GRD03								
		—	—	—	—	—	—	—	—
	DCNTD00								
		—	—	—	—	—	—	—	—

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	DCNTD01								
		—	—	—	—	—	—	—	—
	DCNTD02								
		—	—	—	—	—	—	—	—
	DCNTD03								
		—	—	—	—	—	—	—	—
	TIOR1D1	OSSD13[1:0]		OSSD12[1:0]		OSSD11[1:0]		OSSD10[1:0]	
		IOAD13[1:0]		IOAD12[1:0]		IOAD11[1:0]		IOAD10[1:0]	
	TIOR2D1	—	IOBD13[2:0]			—	IOBD12[2:0]		
		—	IOBD11[2:0]			—	IOBD10[2:0]		
	DSTRD1	—	—	—	—	DSTD13	DSTD12	DSTD11	DSTD10
	DSRD1	—	—	—	—	DSFD13	DSFD12	DSFD11	DSFD10
	DCRD1	—	TRGSELD13[2:0]			—	TRGSELD12[2:0]		
		—	TRGSELD11[2:0]			—	TRGSELD10[2:0]		
	TSRD1	—	—	OVF2D1	OVF1D1	UDFD13	UDFD12	UDFD11	UDFD10
		CMFAD13	CMFAD12	CMFAD11	CMFAD10	CMFBD13	CMFBD12	CMFBD11	CMFBD10
	TIERD1	—	—	OVE2D1	OVE1D1	UDED13	UDED12	UDED11	UDED10
		CMEAD13	CMEAD12	CMEAD11	CMEAD10	CMEBD13	CMEBD12	CMEBD11	CMEBD10
	OCRD10								
		—	—	—	—	—	—	—	—
	OCRD11								
		—	—	—	—	—	—	—	—

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	OCRD12								
		—	—	—	—	—	—	—	—
	OCRD13								
		—	—	—	—	—	—	—	—
	GRD10								
		—	—	—	—	—	—	—	—
	GRD11								
		—	—	—	—	—	—	—	—
	GRD12								
		—	—	—	—	—	—	—	—
	GRD13								
		—	—	—	—	—	—	—	—
	DCNTD10								
		—	—	—	—	—	—	—	—
	DCNTD11								
		—	—	—	—	—	—	—	—

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	DCNTD12								
		—	—	—	—	—	—	—	—
	DCNTD13								
		—	—	—	—	—	—	—	—
	TIOR1D2	OSSD23[1:0]		OSSD22[1:0]		OSSD21[1:0]		OSSD20[1:0]	
		IOAD23[1:0]		IOAD22[1:0]		IOAD21[1:0]		IOAD20[1:0]	
	TIOR2D2	—	IOBD23[2:0]			—	IOBD22[2:0]		
		—	IOBD21[2:0]			—	IOBD20[2:0]		
	DSTRD2	—	—	—	—	DSTD23	DSTD22	DSTD21	DSTD20
	DSRD2	—	—	—	—	DSFD23	DSFD22	DSFD21	DSFD20
	DCRD2	—	TRGSELD23[2:0]			—	TRGSELD22[2:0]		
		—	TRGSELD21[2:0]			—	TRGSELD20[2:0]		
	TSRD2	—	—	OVF2D2	OVF1D2	UDFD23	UDFD22	UDFD21	UDFD20
		CMFAD23	CMFAD22	CMFAD21	CMFAD20	CMFBD23	CMFBD22	CMFBD21	CMFBD20
	TIERD2	—	—	OVE2D2	OVE1D2	UDED23	UDED22	UDED21	UDED20
		CMEAD23	CMEAD22	CMEAD21	CMEAD20	CMEBD23	CMEBD22	CMEBD21	CMEBD20
	OCRD20								
		—	—	—	—	—	—	—	—
	OCRD21								
		—	—	—	—	—	—	—	—
	OCRD22								
		—	—	—	—	—	—	—	—

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	OCRD23								
		—	—	—	—	—	—	—	—
	GRD20								
		—	—	—	—	—	—	—	—
	GRD21								
		—	—	—	—	—	—	—	—
	GRD22								
		—	—	—	—	—	—	—	—
	GRD23								
		—	—	—	—	—	—	—	—
	DCNTD20								
		—	—	—	—	—	—	—	—
	DCNTD21								
		—	—	—	—	—	—	—	—
	DCNTD22								
		—	—	—	—	—	—	—	—

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	DCNTD23								
		—	—	—	—	—	—	—	—
	TIOR1D3	OSSD33[1:0]		OSSD32[1:0]		OSSD31[1:0]		OSSD30[1:0]	
		IOAD33[1:0]		IOAD32[1:0]		IOAD31[1:0]		IOAD30[1:0]	
	TIOR2D3	—	IOBD33[2:0]			—	IOBD32[2:0]		
		—	IOBD31[2:0]			—	IOBD30[2:0]		
	DSTRD3	—	—	—	—	DSTD33	DSTD32	DSTD31	DSTD30
	DSRD3	—	—	—	—	DSFD33	DSFD32	DSFD31	DSFD30
	DCRD3	—	TRGSELD33[2:0]			—	TRGSELD32[2:0]		
		—	TRGSELD31[2:0]			—	TRGSELD30[2:0]		
	TSRD3	—	—	OVF2D3	OVF1D3	UDFD33	UDFD32	UDFD31	UDFD30
		CMFAD33	CMFAD32	CMFAD31	CMFAD30	CMFBD33	CMFBD32	CMFBD31	CMFBD30
	TIERD3	—	—	OVE2D3	OVE1D3	UDED33	UDED32	UDED31	UDED30
		CMEAD33	CMEAD32	CMEAD31	CMEAD30	CMEBD33	CMEBD32	CMEBD31	CMEBD30
	OCRD30								
		—	—	—	—	—	—	—	—
	OCRD31								
		—	—	—	—	—	—	—	—
	OCRD32								
		—	—	—	—	—	—	—	—
	OCRD33								
		—	—	—	—	—	—	—	—

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	GRD30								
		—	—	—	—	—	—	—	—
	GRD31								
		—	—	—	—	—	—	—	—
	GRD32								
		—	—	—	—	—	—	—	—
	GRD33								
		—	—	—	—	—	—	—	—
	DCNTD30								
		—	—	—	—	—	—	—	—
	DCNTD31								
		—	—	—	—	—	—	—	—
	DCNTD32								
		—	—	—	—	—	—	—	—
	DCNTD33								
		—	—	—	—	—	—	—	—

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	TSTRE	—	STRE6	STRE5	STRE4	STRE3	STRE2	STRE1	STRE0
	TCRE0	PSCSEL	—	—	—	—	CKSELE0[2:0]		
	TOCRE0	—	—	—	—	TONEE03	TONEE02	TONEE01	TONEE00
	TIERE0	—	—	—	—	CMEE03	CMEE02	CMEE01	CMEE00
	RLDCRE0	—	—	—	—	RLDENE03	RLDENE02	RLDENE01	RLDENE00
	TSRE0	OVFE03	OVFE02	OVFE01	OVFE00	CMFE03	CMFE02	CMFE01	CMFE00
	PSCRE0	—	—	—	—	—	PSCE0[2:0]		
	SSTRE0	—	—	—	—	SSTRE03	SSTRE02	SSTRE01	SSTRE00
	CYLRE00	CYLRE00[15:8]							
		CYLRE00[7:0]							
	CYLRE01	CYLRE01[15:8]							
		CYLRE01[7:0]							
	CYLRE02	CYLRE02[15:8]							
		CYLRE02[7:0]							
	CYLRE03	CYLRE03[15:8]							
		CYLRE03[7:0]							
	DTRE00	DTRE00[15:8]							
		DTRE00[7:0]							
	DTRE01	DTRE01[15:8]							
		DTRE01[7:0]							
	DTRE02	DTRE02[15:8]							
		DTRE02[7:0]							
	DTRE03	DTRE03[15:8]							
		DTRE03[7:0]							
	CRLDE00	CRLDE00[15:8]							
		CRLDE00[7:0]							
	CRLDE01	CRLDE01[15:8]							
		CRLDE01[7:0]							
	CRLDE02	CRLDE02[15:8]							
		CRLDE02[7:0]							
	CRLDE03	CRLDE03[15:8]							
		CRLDE03[7:0]							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	DRLDE00	DRLDE00[15:8]							
		DRLDE00[7:0]							
	DRLDE01	DRLDE01[15:8]							
		DRLDE01[7:0]							
	DRLDE02	DRLDE02[15:8]							
		DRLDE02[7:0]							
	DRLDE03	DRLDE03[15:8]							
		DRLDE03[7:0]							
	TCNTE00	TCNTE00[15:8]							
		TCNTE00[7:0]							
	TCNTE01	TCNTE01[15:8]							
		TCNTE01[7:0]							
	TCNTE02	TCNTE02[15:8]							
		TCNTE02[7:0]							
	TCNTE03	TCNTE03[15:8]							
		TCNTE03[7:0]							
	PSCCRE00	PSCCE00[7:0]							
	PSCCRE01	PSCCE01[7:0]							
	PSCCRE02	PSCCE02[7:0]							
	PSCCRE03	PSCCE03[7:0]							
	TCRE1	PSCSEL	—	—	—	—	CKSELE1[2:0]		
	TOCRE1	—	—	—	—	TONEE13	TONEE12	TONEE11	TONEE10
	TIERE1	—	—	—	—	CMEE13	CMEE12	CMEE11	CMEE10
	RLDCRE1	—	—	—	—	RLDENE13	RLDENE12	RLDENE11	RLDENE10
	TSRE1	OVFE13	OVFE12	OVFE11	OVFE10	CMFE13	CMFE12	CMFE11	CMFE10
	PSCRE1	—	—	—	—	—	PSCE1[2:0]		
	SSTRE1	—	—	—	—	SSTRE13	SSTRE12	SSTRE11	SSTRE10
	CYLRE10	CYLRE10[15:8]							
		CYLRE10[7:0]							
	CYLRE11	CYLRE11[15:8]							
		CYLRE11[7:0]							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	CYLRE12	CYLRE12[15:8]							
		CYLRE12[7:0]							
	CYLRE13	CYLRE13[15:8]							
		CYLRE13[7:0]							
	DTRE10	DTRE10[15:8]							
		DTRE10[7:0]							
	DTRE11	DTRE11[15:8]							
		DTRE11[7:0]							
	DTRE12	DTRE12[15:8]							
		DTRE12[7:0]							
	DTRE13	DTRE13[15:8]							
		DTRE13[7:0]							
	CRLDE10	CRLDE10[15:8]							
		CRLDE10[7:0]							
	CRLDE11	CRLDE11[15:8]							
		CRLDE11[7:0]							
	CRLDE12	CRLDE12[15:8]							
		CRLDE12[7:0]							
	CRLDE13	CRLDE13[15:8]							
		CRLDE13[7:0]							
	DRLDE10	DRLDE10[15:8]							
		DRLDE10[7:0]							
	DRLDE11	DRLDE11[15:8]							
		DRLDE11[7:0]							
	DRLDE12	DRLDE12[15:8]							
		DRLDE12[7:0]							
	DRLDE13	DRLDE13[15:8]							
		DRLDE13[7:0]							
	TCNTE10	TCNTE10[15:8]							
		TCNTE10[7:0]							
	TCNTE11	TCNTE11[15:8]							
		TCNTE11[7:0]							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	TCNTE12	TCNTE12[15:8]							
		TCNTE12[7:0]							
	TCNTE13	TCNTE13[15:8]							
		TCNTE13[7:0]							
	PSCCRE10	PSCCE10[7:0]							
	PSCCRE11	PSCCE11[7:0]							
	PSCCRE12	PSCCE12[7:0]							
	PSCCRE13	PSCCE13[7:0]							
	TCRE2	PSCSEL	—	—	—	—	CKSELE2[2:0]		
	TOCRE2	—	—	—	—	TONEE23	TONEE22	TONEE21	TONEE20
	TIERE2	—	—	—	—	CMEE23	CMEE22	CMEE21	CMEE20
	RLDCRE2	—	—	—	—	RLDENE23	RLDENE22	RLDENE21	RLDENE20
	TSRE2	OVFE23	OVFE22	OVFE21	OVFE20	CMFE23	CMFE22	CMFE21	CMFE20
	PSCRE2	—	—	—	—	—	PSCE2[2:0]		
	SSTRE2	—	—	—	—	SSTRE23	SSTRE22	SSTRE21	SSTRE20
	CYLRE20	CYLRE20[15:8]							
		CYLRE20[7:0]							
	CYLRE21	CYLRE21[15:8]							
		CYLRE21[7:0]							
	CYLRE22	CYLRE22[15:8]							
		CYLRE22[7:0]							
	CYLRE23	CYLRE23[15:8]							
		CYLRE23[7:0]							
	DTRE20	DTRE20[15:8]							
		DTRE20[7:0]							
	DTRE21	DTRE21[15:8]							
		DTRE21[7:0]							
	DTRE22	DTRE22[15:8]							
		DTRE22[7:0]							
	DTRE23	DTRE23[15:8]							
		DTRE23[7:0]							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	CRLDE20	CRLDE20[15:8]							
		CRLDE20[7:0]							
	CRLDE21	CRLDE21[15:8]							
		CRLDE21[7:0]							
	CRLDE22	CRLDE22[15:8]							
		CRLDE22[7:0]							
	CRLDE23	CRLDE23[15:8]							
		CRLDE23[7:0]							
	DRLDE20	DRLDE20[15:8]							
		DRLDE20[7:0]							
	DRLDE21	DRLDE21[15:8]							
		DRLDE21[7:0]							
	DRLDE22	DRLDE22[15:8]							
		DRLDE22[7:0]							
	DRLDE23	DRLDE23[15:8]							
		DRLDE23[7:0]							
	TCNTE20	TCNTE20[15:8]							
		TCNTE20[7:0]							
	TCNTE21	TCNTE21[15:8]							
		TCNTE21[7:0]							
	TCNTE22	TCNTE22[15:8]							
		TCNTE22[7:0]							
	TCNTE23	TCNTE23[15:8]							
		TCNTE23[7:0]							
	PSCCRE20	PSCCE20[7:0]							
	PSCCRE21	PSCCE21[7:0]							
	PSCCRE22	PSCCE22[7:0]							
	PSCCRE23	PSCCE23[7:0]							
	TCRE3	PSCSEL	—	—	—	—	CKSELE3[2:0]		

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	TOCRE3	—	—	—	—	TONEE33	TONEE32	TONEE31	TONEE30
	TIERE3	—	—	—	—	CMEE33	CMEE32	CMEE31	CMEE30
	RLDCRE3	—	—	—	—	RLDENE33	RLDENE32	RLDENE31	RLDENE30
	TSRE3	OVFE33	OVFE32	OVFE31	OVFE30	CMFE33	CMFE32	CMFE31	CMFE30
	PSCRE3	—	—	—	—	—	PSCE3[2:0]		
	SSTRE3	—	—	—	—	SSTRE33	SSTRE32	SSTRE31	SSTRE30
	CYLRE30	CYLRE30[15:8]							
		CYLRE30[7:0]							
	CYLRE31	CYLRE31[15:8]							
		CYLRE31[7:0]							
	CYLRE32	CYLRE32[15:8]							
		CYLRE32[7:0]							
	CYLRE33	CYLRE33[15:8]							
		CYLRE33[7:0]							
	DTRE30	DTRE30[15:8]							
		DTRE30[7:0]							
	DTRE31	DTRE31[15:8]							
		DTRE31[7:0]							
	DTRE32	DTRE32[15:8]							
		DTRE32[7:0]							
	DTRE33	DTRE33[15:8]							
		DTRE33[7:0]							
	CRLDE30	CRLDE30[15:8]							
		CRLDE30[7:0]							
	CRLDE31	CRLDE31[15:8]							
		CRLDE31[7:0]							
	CRLDE32	CRLDE32[15:8]							
		CRLDE32[7:0]							
	CRLDE33	CRLDE33[15:8]							
		CRLDE33[7:0]							
	DRLDE30	DRLDE30[15:8]							
		DRLDE30[7:0]							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	DRLDE31	DRLDE31[15:8]							
		DRLDE31[7:0]							
	DRLDE32	DRLDE32[15:8]							
		DRLDE32[7:0]							
	DRLDE33	DRLDE33[15:8]							
		DRLDE33[7:0]							
	TCNTE30	TCNTE30[15:8]							
		TCNTE30[7:0]							
	TCNTE31	TCNTE31[15:8]							
		TCNTE31[7:0]							
	TCNTE32	TCNTE32[15:8]							
		TCNTE32[7:0]							
	TCNTE33	TCNTE33[15:8]							
		TCNTE33[7:0]							
	PSCCRE30	PSCCE30[7:0]							
	PSCCRE31	PSCCE31[7:0]							
	PSCCRE32	PSCCE32[7:0]							
	PSCCRE33	PSCCE33[7:0]							
	TCRE4	PSCSEL	—	—	—	—	CKSELE4[2:0]		
	TOCRE4	—	—	—	—	TONEE43	TONEE42	TONEE41	TONEE40
	TIERE4	—	—	—	—	CMEE43	CMEE42	CMEE41	CMEE40
	RLDCRE4	—	—	—	—	RLDENE43	RLDENE42	RLDENE41	RLDENE40
	TSRE4	OVFE43	OVFE42	OVFE41	OVFE40	CMFE43	CMFE42	CMFE41	CMFE40
	PSCRE4	—	—	—	—	—	PSCE4[2:0]		
	SSTRE4	—	—	—	—	SSTRE43	SSTRE42	SSTRE41	SSTRE40
	CYLRE40	CYLRE40[15:8]							
		CYLRE40[7:0]							
	CYLRE41	CYLRE41[15:8]							
		CYLRE41[7:0]							
	CYLRE42	CYLRE42[15:8]							
		CYLRE42[7:0]							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	CYLRE43	CYLRE43[15:8]							
		CYLRE43[7:0]							
	DTRE40	DTRE40[15:8]							
		DTRE40[7:0]							
	DTRE41	DTRE41[15:8]							
		DTRE41[7:0]							
	DTRE42	DTRE42[15:8]							
		DTRE42[7:0]							
	DTRE43	DTRE43[15:8]							
		DTRE43[7:0]							
	CRLDE40	CRLDE40[15:8]							
		CRLDE40[7:0]							
	CRLDE41	CRLDE41[15:8]							
		CRLDE41[7:0]							
	CRLDE42	CRLDE42[15:8]							
		CRLDE42[7:0]							
	CRLDE43	CRLDE43[15:8]							
		CRLDE43[7:0]							
	DRLDE40	DRLDE40[15:8]							
		DRLDE40[7:0]							
	DRLDE41	DRLDE41[15:8]							
		DRLDE41[7:0]							
	DRLDE42	DRLDE42[15:8]							
		DRLDE42[7:0]							
	DRLDE43	DRLDE43[15:8]							
		DRLDE43[7:0]							
	TCNTE40	TCNTE40[15:8]							
		TCNTE40[7:0]							
	TCNTE41	TCNTE41[15:8]							
		TCNTE41[7:0]							
	TCNTE42	TCNTE42[15:8]							
		TCNTE42[7:0]							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	TCNTE43	TCNTE43[15:8]							
		TCNTE43[7:0]							
	PSCCRE40	PSCCE40[7:0]							
	PSCCRE41	PSCCE41[7:0]							
	PSCCRE42	PSCCE42[7:0]							
	PSCCRE43	PSCCE43[7:0]							
	TCRE5	PSCSEL	—	—	—	—	CKSELE5[2:0]		
	TOCRE5	—	—	—	—	TONEE53	TONEE52	TONEE51	TONEE50
	TIERE5	—	—	—	—	CMEE53	CMEE52	CMEE51	CMEE50
	RLDCRE5	—	—	—	—	RLDENE53	RLDENE52	RLDENE51	RLDENE50
	TSRE5	OVFE53	OVFE52	OVFE51	OVFE50	CMFE53	CMFE52	CMFE51	CMFE50
	PSCRE5	—	—	—	—	—	PSCE5[2:0]		
	SSTRE5	—	—	—	—	SSTRE53	SSTRE52	SSTRE51	SSTRE50
	CYLRE50	CYLRE50[15:8]							
		CYLRE50[7:0]							
	CYLRE51	CYLRE51[15:8]							
		CYLRE51[7:0]							
	CYLRE52	CYLRE52[15:8]							
		CYLRE52[7:0]							
	CYLRE53	CYLRE53[15:8]							
		CYLRE53[7:0]							
	DTRE50	DTRE50[15:8]							
		DTRE50[7:0]							
	DTRE51	DTRE51[15:8]							
		DTRE51[7:0]							
	DTRE52	DTRE52[15:8]							
		DTRE52[7:0]							
	DTRE53	DTRE53[15:8]							
		DTRE53[7:0]							
	CRLDE50	CRLDE50[15:8]							
		CRLDE50[7:0]							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	CRLDE51	CRLDE51[15:8]							
		CRLDE51[7:0]							
	CRLDE52	CRLDE52[15:8]							
		CRLDE52[7:0]							
	CRLDE53	CRLDE53[15:8]							
		CRLDE53[7:0]							
	DRLDE50	DRLDE50[15:8]							
		DRLDE50[7:0]							
	DRLDE51	DRLDE51[15:8]							
		DRLDE51[7:0]							
	DRLDE52	DRLDE52[15:8]							
		DRLDE52[7:0]							
	DRLDE53	DRLDE53[15:8]							
		DRLDE53[7:0]							
	TCNTE50	TCNTE50[15:8]							
		TCNTE50[7:0]							
	TCNTE51	TCNTE51[15:8]							
		TCNTE51[7:0]							
	TCNTE52	TCNTE52[15:8]							
		TCNTE52[7:0]							
	TCNTE53	TCNTE53[15:8]							
		TCNTE53[7:0]							
	PSCCRE50	PSCCE50[7:0]							
	PSCCRE51	PSCCE51[7:0]							
	PSCCRE52	PSCCE52[7:0]							
	PSCCRE53	PSCCE53[7:0]							
	TCRE6	PSCSEL	—	—	—	—	CKSELE6[2:0]		
	TOCRE6	—	—	—	—	TONEE63	TONEE62	TONEE61	TONEE60
	TIERE6	—	—	—	—	CMEE63	CMEE62	CMEE61	CMEE60
	RLDCRE6	—	—	—	—	RLDENE63	RLDENE62	RLDENE61	RLDENE60
	TSRE6	OVFE63	OVFE62	OVFE61	OVFE60	CMFE63	CMFE62	CMFE61	CMFE60
	PSCRE6	—	—	—	—	—	PSCE6[2:0]		

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	SSTRE6	—	—	—	—	SSTRE63	SSTRE62	SSTRE61	SSTRE60
	CYLRE60	CYLRE60[15:8]							
		CYLRE60[7:0]							
	CYLRE61	CYLRE61[15:8]							
		CYLRE61[7:0]							
	CYLRE62	CYLRE62[15:8]							
		CYLRE62[7:0]							
	CYLRE63	CYLRE63[15:8]							
		CYLRE63[7:0]							
	DTRE60	DTRE60[15:8]							
		DTRE60[7:0]							
	DTRE61	DTRE61[15:8]							
		DTRE61[7:0]							
	DTRE62	DTRE62[15:8]							
		DTRE62[7:0]							
	DTRE63	DTRE63[15:8]							
		DTRE63[7:0]							
	CRLDE60	CRLDE60[15:8]							
		CRLDE60[7:0]							
	CRLDE61	CRLDE61[15:8]							
		CRLDE61[7:0]							
	CRLDE62	CRLDE62[15:8]							
		CRLDE62[7:0]							
	CRLDE63	CRLDE63[15:8]							
		CRLDE63[7:0]							
	DRLDE60	DRLDE60[15:8]							
		DRLDE60[7:0]							
	DRLDE61	DRLDE61[15:8]							
		DRLDE61[7:0]							
	DRLDE62	DRLDE62[15:8]							
		DRLDE62[7:0]							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	DRLDE63	DRLDE63[15:8]							
		DRLDE63[7:0]							
	TCNTE60	TCNTE60[15:8]							
		TCNTE60[7:0]							
	TCNTE61	TCNTE61[15:8]							
		TCNTE61[7:0]							
	TCNTE62	TCNTE62[15:8]							
		TCNTE62[7:0]							
	TCNTE63	TCNTE63[15:8]							
		TCNTE63[7:0]							
	PSCCRE60	PSCCE60[7:0]							
	PSCCRE61	PSCCE61[7:0]							
	PSCCRE62	PSCCE62[7:0]							
	PSCCRE63	PSCCE63[7:0]							
	TSTRF	—	—	—	—	STRF27	STRF26	STRF25	STRF24
		STRF23	STRF22	STRF21	STRF20	STRF19	STRF18	STRF17	STRF16
		STRF15	STRF14	STRF13	STRF12	STRF11	STRF10	STRF9	STRF8
		STRF7	STRF6	STRF5	STRF4	STRF3	STRF2	STRF1	STRF0
	NCCRF	—	—	—	—	NCEF27	NCEF26	NCEF25	NCEF24
		NCEF23	NCEF22	NCEF21	NCEF20	NCEF19	NCEF18	NCEF17	NCEF16
		NCEF15	NCEF14	NCEF13	NCEF12	NCEF11	NCEF10	NCEF9	NCEF8
		NCEF7	NCEF6	NCEF5	NCEF4	NCEF3	NCEF2	NCEF1	NCEF0
	NCMCR1F	—	—	—	—	NCM1F27	NCM1F26	NCM1F25	NCM1F24
		NCM1F23	NCM1F22	NCM1F21	NCM1F20	NCM1F19	NCM1F18	NCM1F17	NCM1F16
		NCM1F15	NCM1F14	NCM1F13	NCM1F12	NCM1F11	NCM1F10	NCM1F9	NCM1F8
		NCM1F7	NCM1F6	NCM1F5	NCM1F4	NCM1F3	NCM1F2	NCM1F1	NCM1F0
	NCNTFA0	NCNTFA0[7:0]							
	NCRFA0	NCTFA0[7:0]							
	NCNTFA1	NCNTFA1[7:0]							
	NCRFA1	NCTFA1[7:0]							
	NCNTFA2	NCNTFA2[7:0]							
	NCRFA2	NCTFA2[7:0]							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	NCNTFA3	NCNTFA3[7:0]							
	NCRFA3	NCTFA3[7:0]							
	NCNTFA4	NCNTFA4[7:0]							
	NCRFA4	NCTFA4[7:0]							
	NCNTFA5	NCNTFA5[7:0]							
	NCRFA5	NCTFA5[7:0]							
	NCNTFA6	NCNTFA6[7:0]							
	NCRFA6	NCTFA6[7:0]							
	NCNTFA7	NCNTFA7[7:0]							
	NCRFA7	NCTFA7[7:0]							
	NCNTFA8	NCNTFA8[7:0]							
	NCRFA8	NCTFA8[7:0]							
	NCNTFA9	NCNTFA9[7:0]							
	NCRFA9	NCTFA9[7:0]							
	NCNTFA10	NCNTFA10[7:0]							
	NCRFA10	NCTFA10[7:0]							
	NCNTFA11	NCNTFA11[7:0]							
	NCRFA11	NCTFA11[7:0]							
	NCNTFA12	NCNTFA12[7:0]							
	NCRFA12	NCTFA12[7:0]							
	NCNTFA13	NCNTFA13[7:0]							
	NCRFA13	NCTFA13[7:0]							
	NCNTFA14	NCNTFA14[7:0]							
	NCRFA14	NCTFA14[7:0]							
	NCNTFA15	NCNTFA15[7:0]							
	NCRFA15	NCTFA15[7:0]							
	NCNTFA16	NCNTFA16[7:0]							
	NCRFA16	NCTFA16[7:0]							
	NCNTFA17	NCNTFA17[7:0]							
	NCRFA17	NCTFA17[7:0]							
	NCNTFA18	NCNTFA18[7:0]							
	NCRFA18	NCTFA18[7:0]							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	NCNTFA19	NCNTFA19[7:0]							
	NCRFA19	NCTFA19[7:0]							
	NCNTFA20	NCNTFA20[7:0]							
	NCRFA20	NCTFA20[7:0]							
	NCNTFA21	NCNTFA21[7:0]							
	NCRFA21	NCTFA21[7:0]							
	NCNTFA22	NCNTFA22[7:0]							
	NCRFA22	NCTFA22[7:0]							
	NCNTFA23	NCNTFA23[7:0]							
	NCRFA23	NCTFA23[7:0]							
	NCNTFA24	NCNTFA24[7:0]							
	NCRFA24	NCTFA24[7:0]							
	NCNTFA25	NCNTFA25[7:0]							
	NCRFA25	NCTFA25[7:0]							
	NCNTFA26	NCNTFA26[7:0]							
	NCRFA26	NCTFA26[7:0]							
	NCNTFA27	NCNTFA27[7:0]							
	NCRFA27	NCTFA27[7:0]							
	NCNTFB0	NCNTFB0[7:0]							
	NCRFB0	NCTFB0[7:0]							
	NCNTFB1	NCNTFB1[7:0]							
	NCRFB1	NCTFB1[7:0]							
	NCNTFB2	NCNTFB2[7:0]							
	NCRFB2	NCTFB2[7:0]							
	TCRF0	CKSELF0[2:0]			MDF0[2:0]			EGSELF0[1:0]	
	TIERF0	—	—	—	—	OVECF0	OVEBF0	OVEAF0	ICEF0
	TSRF0	—	—	—	—	OVFCF0	OVFBF0	OVFAF0	ICFF0
	ECNTAF0	ECNTAF0[23:16]							
		ECNTAF0[15:8]							
		ECNTAF0[7:0]							
		—	—	—	—	—	—	—	—

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	ECNTBF0	ECNTBF0[15:8]							
		ECNTBF0[7:0]							
	GRBF0	GRBF0[15:8]							
		GRBF0[7:0]							
	ECNTCF0	ECNTCF0[23:16]							
		ECNTCF0[15:8]							
		ECNTCF0[7:0]							
		—	—	—	—	—	—	—	—
	GRAF0	GRAF0[23:16]							
		GRAF0[15:8]							
		GRAF0[7:0]							
		—	—	—	—	—	—	—	—
	CDRF0	CDRF0[23:16]							
		CDRF0[15:8]							
		CDRF0[7:0]							
		—	—	—	—	—	—	—	—
	GRCF0	GRCF0[23:16]							
		GRCF0[15:8]							
		GRCF0[7:0]							
		—	—	—	—	—	—	—	—
	TCRF1	CKSELF1[2:0]			MDF1[2:0]			EGSELF1[1:0]	
	TIERF1	—	—	—	—	OVECF1	OVEBF1	OVEAF1	ICEF1
	TSRF1	—	—	—	—	OVFCF1	OVFBF1	OVFAF1	ICFF1
	ECNTAF1	ECNTAF1[23:16]							
		ECNTAF1[15:8]							
		ECNTAF1[7:0]							
		—	—	—	—	—	—	—	—
	ECNTBF1	ECNTBF1[15:8]							
		ECNTBF1[7:0]							
	GRBF1	GRBF1[15:8]							
		GRBF1[7:0]							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	ECNTCF1	ECNTCF1[23:16]							
		ECNTCF1[15:8]							
		ECNTCF1[7:0]							
		—	—	—	—	—	—	—	—
	GRAF1	GRAF1[23:16]							
		GRAF1[15:8]							
		GRAF1[7:0]							
		—	—	—	—	—	—	—	—
	CDRF1	CDRF1[23:16]							
		CDRF1[15:8]							
		CDRF1[7:0]							
		—	—	—	—	—	—	—	—
	GRCF1	GRCF1[23:16]							
		GRCF1[15:8]							
		GRCF1[7:0]							
		—	—	—	—	—	—	—	—
	TCRF2	CKSELF2[2:0]			MDF2[2:0]			EGSELF2[1:0]	
	TIERF2	—	—	—	—	OVECF2	OVEBF2	OVEAF2	ICEF2
	TSRF2	—	—	—	—	OVFCF2	OVFBF2	OVFAF2	ICFF2
	ECNTAF2	ECNTAF2[23:16]							
		ECNTAF2[15:8]							
		ECNTAF2[7:0]							
		—	—	—	—	—	—	—	—
	ECNTBF2	ECNTBF2[15:8]							
		ECNTBF2[7:0]							
	GRBF2	GRBF2[15:8]							
		GRBF2[7:0]							
	ECNTCF2	ECNTCF2[23:16]							
		ECNTCF2[15:8]							
		ECNTCF2[7:0]							
		—	—	—	—	—	—	—	—

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	GRAF2	GRAF2[23:16]							
		GRAF2[15:8]							
		GRAF2[7:0]							
		—	—	—	—	—	—	—	—
	CDRF2	CDRF2[23:16]							
		CDRF2[15:8]							
		CDRF2[7:0]							
		—	—	—	—	—	—	—	—
	GRCF2	GRCF2[23:16]							
		GRCF2[15:8]							
		GRCF2[7:0]							
		—	—	—	—	—	—	—	—
	TCRF3	CKSELF3[2:0]			MDF3[2:0]			EGSELF3[1:0]	
	TIERF3	—	—	—	—	OVECF3	OVEBF3	OVEAF3	ICEF3
	TSRF3	—	—	—	—	OVFCF3	OVFBF3	OVFAF3	ICFF3
	ECNTAF3	ECNTAF3[23:16]							
		ECNTAF3[15:8]							
		ECNTAF3[7:0]							
		—	—	—	—	—	—	—	—
	ECNTBF3	ECNTBF3[15:8]							
		ECNTBF3[7:0]							
	GRBF3	GRBF3[15:8]							
		GRBF3[7:0]							
	ECNTCF3	ECNTCF3[23:16]							
		ECNTCF3[15:8]							
		ECNTCF3[7:0]							
		—	—	—	—	—	—	—	—
	GRAF3	GRAF3[23:16]							
		GRAF3[15:8]							
		GRAF3[7:0]							
		—	—	—	—	—	—	—	—

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	CDRF3	CDRF3[23:16]							
		CDRF3[15:8]							
		CDRF3[7:0]							
		—	—	—	—	—	—	—	—
	GRCF3	GRCF3[23:16]							
		GRCF3[15:8]							
		GRCF3[7:0]							
		—	—	—	—	—	—	—	—
	TCRF4	CKSELF4[2:0]			MDF4[2:0]			EGSELF4[1:0]	
	TIERF4	—	—	—	—	OVECF4	OVEBF4	OVEAF4	ICEF4
	TSRF4	—	—	—	—	OVFCF4	OVFBF4	OVFAF4	ICFF4
	ECNTAF4	ECNTAF4[23:16]							
		ECNTAF4[15:8]							
		ECNTAF4[7:0]							
		—	—	—	—	—	—	—	—
	ECNTBF4	ECNTBF4[15:8]							
		ECNTBF4[7:0]							
	GRBF4	GRBF4[15:8]							
		GRBF4[7:0]							
	ECNTCF4	ECNTCF4[23:16]							
		ECNTCF4[15:8]							
		ECNTCF4[7:0]							
		—	—	—	—	—	—	—	—
	GRAF4	GRAF4[23:16]							
		GRAF4[15:8]							
		GRAF4[7:0]							
		—	—	—	—	—	—	—	—
	CDRF4	CDRF4[23:16]							
		CDRF4[15:8]							
		CDRF4[7:0]							
		—	—	—	—	—	—	—	—

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	GRCF4	GRCF4[23:16]							
		GRCF4[15:8]							
		GRCF4[7:0]							
		—	—	—	—	—	—	—	—
	TCRF5	CKSELF5[2:0]			MDF5[2:0]			EGSELF5[1:0]	
	TIERF5	—	—	—	—	OVECF5	OVEBF5	OVEAF5	ICEF5
	TSRF5	—	—	—	—	OVFCF5	OVFBF5	OVFAF5	ICFF5
	ECNTAF5	ECNTAF5[23:16]							
		ECNTAF5[15:8]							
		ECNTAF5[7:0]							
		—	—	—	—	—	—	—	—
	ECNTBF5	ECNTBF5[15:8]							
		ECNTBF5[7:0]							
	GRBF5	GRBF5[15:8]							
		GRBF5[7:0]							
	ECNTCF5	ECNTCF5[23:16]							
		ECNTCF5[15:8]							
		ECNTCF5[7:0]							
		—	—	—	—	—	—	—	—
	GRAF5	GRAF5[23:16]							
		GRAF5[15:8]							
		GRAF5[7:0]							
		—	—	—	—	—	—	—	—
	CDRF5	CDRF5[23:16]							
		CDRF5[15:8]							
		CDRF5[7:0]							
		—	—	—	—	—	—	—	—
	GRCF5	GRCF5[23:16]							
		GRCF5[15:8]							
		GRCF5[7:0]							
		—	—	—	—	—	—	—	—

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	TCRF6	CKSELF6[2:0]			MDF6[2:0]			EGSELF6[1:0]	
	TIERF6	—	—	—	—	OVECF6	OVEBF6	OVEAF6	ICEF6
	TSRF6	—	—	—	—	OVFCF6	OVFBF6	OVFAF6	ICFF6
	ECNTAF6	ECNTAF6[23:16]							
		ECNTAF6[15:8]							
		ECNTAF6[7:0]							
		—	—	—	—	—	—	—	—
	ECNTBF6	ECNTBF6[15:8]							
		ECNTBF6[7:0]							
	GRBF6	GRBF6[15:8]							
		GRBF6[7:0]							
	ECNTCF6	ECNTCF6[23:16]							
		ECNTCF6[15:8]							
		ECNTCF6[7:0]							
		—	—	—	—	—	—	—	—
	GRAF6	GRAF6[23:16]							
		GRAF6[15:8]							
		GRAF6[7:0]							
		—	—	—	—	—	—	—	—
	CDRF6	CDRF6[23:16]							
		CDRF6[15:8]							
		CDRF6[7:0]							
		—	—	—	—	—	—	—	—
	GRCF6	GRCF6[23:16]							
		GRCF6[15:8]							
		GRCF6[7:0]							
		—	—	—	—	—	—	—	—
	TCRF7	CKSELF7[2:0]			MDF7[2:0]			EGSELF7[1:0]	
	TIERF7	—	—	—	—	OVECF7	OVEBF7	OVEAF7	ICEF7
	TSRF7	—	—	—	—	OVFCF7	OVFBF7	OVFAF7	ICFF7

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	ECNTAF7	ECNTAF7[23:16]							
		ECNTAF7[15:8]							
		ECNTAF7[7:0]							
		—	—	—	—	—	—	—	—
	ECNTBF7	ECNTBF7[15:8]							
		ECNTBF7[7:0]							
	GRBF7	GRBF7[15:8]							
		GRBF7[7:0]							
	ECNTCF7	ECNTCF7[23:16]							
		ECNTCF7[15:8]							
		ECNTCF7[7:0]							
		—	—	—	—	—	—	—	—
	GRAF7	GRAF7[23:16]							
		GRAF7[15:8]							
		GRAF7[7:0]							
		—	—	—	—	—	—	—	—
	CDRF7	CDRF7[23:16]							
		CDRF7[15:8]							
		CDRF7[7:0]							
		—	—	—	—	—	—	—	—
	GRCF7	GRCF7[23:16]							
		GRCF7[15:8]							
		GRCF7[7:0]							
		—	—	—	—	—	—	—	—
	TCRF8	CKSELF8[2:0]			MDF8[2:0]			EGSELF8[1:0]	
	TIERF8	—	—	—	—	OVECF8	OVEBF8	OVEAF8	ICEF8
	TSRF8	—	—	—	—	OVFCF8	OVFBF8	OVFAF8	ICFF8
	ECNTAF8	ECNTAF8[23:16]							
		ECNTAF8[15:8]							
		ECNTAF8[7:0]							
		—	—	—	—	—	—	—	—

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	ECNTBF8	ECNTBF8[15:8]							
		ECNTBF8[7:0]							
	GRBF8	GRBF8[15:8]							
		GRBF8[7:0]							
	ECNTCF8	ECNTCF8[23:16]							
		ECNTCF8[15:8]							
		ECNTCF8[7:0]							
		—	—	—	—	—	—	—	—
	GRAF8	GRAF8[23:16]							
		GRAF8[15:8]							
		GRAF8[7:0]							
		—	—	—	—	—	—	—	—
	CDRF8	CDRF8[23:16]							
		CDRF8[15:8]							
		CDRF8[7:0]							
		—	—	—	—	—	—	—	—
	GRCF8	GRCF8[23:16]							
		GRCF8[15:8]							
		GRCF8[7:0]							
		—	—	—	—	—	—	—	—
	TCRF9	CKSELF9[2:0]			MDF9[2:0]			EGSELF9[1:0]	
	TIERF9	—	—	—	—	OVECF9	OVEBF9	OVEAF9	ICEF9
	TSRF9	—	—	—	—	OVFCF9	OVFBF9	OVFAF9	ICFF9
	ECNTAF9	ECNTAF9[23:16]							
		ECNTAF9[15:8]							
		ECNTAF9[7:0]							
		—	—	—	—	—	—	—	—
	ECNTBF9	ECNTBF9[15:8]							
		ECNTBF9[7:0]							
	GRBF9	GRBF9[15:8]							
		GRBF9[7:0]							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	ECNTCF9	ECNTCF9[23:16]							
		ECNTCF9[15:8]							
		ECNTCF9[7:0]							
		—	—	—	—	—	—	—	—
	GRAF9	GRAF9[23:16]							
		GRAF9[15:8]							
		GRAF9[7:0]							
		—	—	—	—	—	—	—	—
	CDRF9	CDRF9[23:16]							
		CDRF9[15:8]							
		CDRF9[7:0]							
		—	—	—	—	—	—	—	—
	GRCF9	GRCF9[23:16]							
		GRCF9[15:8]							
		GRCF9[7:0]							
		—	—	—	—	—	—	—	—
	TCRF10	CKSELF10[2:0]				MDF10[2:0]		EGSELF10[1:0]	
	TIERF10	—	—	—	—	OVECF10	OVEBF10	OVEAF10	ICEF10
	TSRF10	—	—	—	—	OVFCF10	OVFBF10	OVFAF10	ICFF10
	ECNTAF10	ECNTAF10[23:16]							
		ECNTAF10[15:8]							
		ECNTAF10[7:0]							
		—	—	—	—	—	—	—	—
	ECNTBF10	ECNTBF10[15:8]							
		ECNTBF10[7:0]							
	GRBF10	GRBF10[15:8]							
		GRBF10[7:0]							
	ECNTCF10	ECNTCF10[23:16]							
		ECNTCF10[15:8]							
		ECNTCF10[7:0]							
		—	—	—	—	—	—	—	—

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	GRAF10	GRAF10[23:16]							
		GRAF10[15:8]							
		GRAF10[7:0]							
		—	—	—	—	—	—	—	—
	CDRF10	CDRF10[23:16]							
		CDRF10[15:8]							
		CDRF10[7:0]							
		—	—	—	—	—	—	—	—
	GRCF10	GRCF10[23:16]							
		GRCF10[15:8]							
		GRCF10[7:0]							
		—	—	—	—	—	—	—	—
	TCRF11	CKSELF11[2:0]			MDF11[2:0]			EGSELF11[1:0]	
	TIERF11	—	—	—	—	OVECF11	OVEBF11	OVEAF11	ICEF11
	TSRF11	—	—	—	—	OVFCF11	OVFBF11	OVFAF11	ICFF11
	ECNTAF11	ECNTAF11[23:16]							
		ECNTAF11[15:8]							
		ECNTAF11[7:0]							
		—	—	—	—	—	—	—	—
	ECNTBF11	ECNTBF11[15:8]							
		ECNTBF11[7:0]							
	GRBF11	GRBF11[15:8]							
		GRBF11[7:0]							
	ECNTCF11	ECNTCF11[23:16]							
		ECNTCF11[15:8]							
		ECNTCF11[7:0]							
		—	—	—	—	—	—	—	—
	GRAF11	GRAF11[23:16]							
		GRAF11[15:8]							
		GRAF11[7:0]							
		—	—	—	—	—	—	—	—

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	CDRF11	CDRF11[23:16]							
		CDRF11[15:8]							
		CDRF11[7:0]							
		—	—	—	—	—	—	—	—
	GRCF11	GRCF11[23:16]							
		GRCF11[15:8]							
		GRCF11[7:0]							
		—	—	—	—	—	—	—	—
	TCRF12	CKSELF12[2:0]			MDF12[2:0]			EGSELF12[1:0]	
	TIERF12	—	—	—	—	OVECF12	OVEBF12	OVEAF12	ICEF12
	TSRF12	—	—	—	—	OVFCF12	OVFBF12	OVFAF12	ICFF12
	ECNTAF12	ECNTAF12[23:16]							
		ECNTAF12[15:8]							
		ECNTAF12[7:0]							
		—	—	—	—	—	—	—	—
	ECNTBF12	ECNTBF12[15:8]							
		ECNTBF12[7:0]							
	GRBF12	GRBF12[15:8]							
		GRBF12[7:0]							
	ECNTCF12	ECNTCF12[23:16]							
		ECNTCF12[15:8]							
		ECNTCF12[7:0]							
		—	—	—	—	—	—	—	—
	GRAF12	GRAF12[23:16]							
		GRAF12[15:8]							
		GRAF12[7:0]							
		—	—	—	—	—	—	—	—
	CDRF12	CDRF12[23:16]							
		CDRF12[15:8]							
		CDRF12[7:0]							
		—	—	—	—	—	—	—	—

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	GRCF12	GRCF12[23:16]							
		GRCF12[15:8]							
		GRCF12[7:0]							
		—	—	—	—	—	—	—	—
	GRDF12	GRDF12[23:16]							
		GRDF12[15:8]							
		GRDF12[7:0]							
		—	—	—	—	—	—	—	—
	TCRF13	CKSELF13[2:0]			MDF13[2:0]			EGSELF13[1:0]	
	TIERF13	—	—	—	—	OVECF13	OVEBF13	OVEAF13	ICEF13
	TSRF13	—	—	—	—	OVFCF13	OVFBF13	OVFAF13	ICFF13
	ECNTAF13	ECNTAF13[23:16]							
		ECNTAF13[15:8]							
		ECNTAF13[7:0]							
		—	—	—	—	—	—	—	—
	ECNTBF13	ECNTBF13[15:8]							
		ECNTBF13[7:0]							
	GRBF13	GRBF13[15:8]							
		GRBF13[7:0]							
	ECNTCF13	ECNTCF13[23:16]							
		ECNTCF13[15:8]							
		ECNTCF13[7:0]							
		—	—	—	—	—	—	—	—
	GRAF13	GRAF13[23:16]							
		GRAF13[15:8]							
		GRAF13[7:0]							
		—	—	—	—	—	—	—	—
	CDRF13	CDRF13[23:16]							
		CDRF13[15:8]							
		CDRF13[7:0]							
		—	—	—	—	—	—	—	—

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	GRCF13	GRCF13[23:16]							
		GRCF13[15:8]							
		GRCF13[7:0]							
		—	—	—	—	—	—	—	—
	GRDF13	GRDF13[23:16]							
		GRDF13[15:8]							
		GRDF13[7:0]							
		—	—	—	—	—	—	—	—
	TCRF14	CKSELF14[2:0]			MDF14[2:0]			EGSELF14[1:0]	
	TIERF14	—	—	—	—	OVECF14	OVEBF14	OVEAF14	ICEF14
	TSRF14	—	—	—	—	OVFCF14	OVFBF14	OVFAF14	ICFF14
	ECNTAF14	ECNTAF14[23:16]							
		ECNTAF14[15:8]							
		ECNTAF14[7:0]							
		—	—	—	—	—	—	—	—
	ECNTBF14	ECNTBF14[15:8]							
		ECNTBF14[7:0]							
	GRBF14	GRBF14[15:8]							
		GRBF14[7:0]							
	ECNTCF14	ECNTCF14[23:16]							
		ECNTCF14[15:8]							
		ECNTCF14[7:0]							
		—	—	—	—	—	—	—	—
	GRAF14	GRAF14[23:16]							
		GRAF14[15:8]							
		GRAF14[7:0]							
		—	—	—	—	—	—	—	—
	CDRF14	CDRF14[23:16]							
		CDRF14[15:8]							
		CDRF14[7:0]							
		—	—	—	—	—	—	—	—

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	GRCF14	GRCF14[23:16]							
		GRCF14[15:8]							
		GRCF14[7:0]							
		—	—	—	—	—	—	—	—
	GRDF14	GRDF14[23:16]							
		GRDF14[15:8]							
		GRDF14[7:0]							
		—	—	—	—	—	—	—	—
	TCRF15	CKSELF15[2:0]			MDF15[2:0]			EGSELF15[1:0]	
	TIERF15	—	—	—	—	OVECF15	OVEBF15	OVEAF15	ICEF15
	TSRF15	—	—	—	—	OVFCF15	OVFBF15	OVFAF15	ICFF15
	ECNTAF15	ECNTAF15[23:16]							
		ECNTAF15[15:8]							
		ECNTAF15[7:0]							
		—	—	—	—	—	—	—	—
	ECNTBF15	ECNTBF15[15:8]							
		ECNTBF15[7:0]							
	GRBF15	GRBF15[15:8]							
		GRBF15[7:0]							
	ECNTCF15	ECNTCF15[23:16]							
		ECNTCF15[15:8]							
		ECNTCF15[7:0]							
		—	—	—	—	—	—	—	—
	GRAF15	GRAF15[23:16]							
		GRAF15[15:8]							
		GRAF15[7:0]							
		—	—	—	—	—	—	—	—
	CDRF15	CDRF15[23:16]							
		CDRF15[15:8]							
		CDRF15[7:0]							
		—	—	—	—	—	—	—	—

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	GRCF15	GRCF15[23:16]							
		GRCF15[15:8]							
		GRCF15[7:0]							
		—	—	—	—	—	—	—	—
	GRDF15	GRDF15[23:16]							
		GRDF15[15:8]							
		GRDF15[7:0]							
		—	—	—	—	—	—	—	—
	TCRF16	CKSELF16[2:0]			MDF16[2:0]			EGSELF16[1:0]	
	TIERF16	—	—	—	—	OVECF16	OVEBF16	OVEAF16	ICEF16
	TSRF16	—	—	—	—	OVFCF16	OVFBF16	OVFAF16	ICFF16
	ECNTAF16	ECNTAF16[23:16]							
		ECNTAF16[15:8]							
		ECNTAF16[7:0]							
		—	—	—	—	—	—	—	—
	ECNTBF16	ECNTBF16[15:8]							
		ECNTBF16[7:0]							
	GRBF16	GRBF16[15:8]							
		GRBF16[7:0]							
	ECNTCF16	ECNTCF16[23:16]							
		ECNTCF16[15:8]							
		ECNTCF16[7:0]							
		—	—	—	—	—	—	—	—
	GRAF16	GRAF16[23:16]							
		GRAF16[15:8]							
		GRAF16[7:0]							
		—	—	—	—	—	—	—	—
	CDRF16	CDRF16[23:16]							
		CDRF16[15:8]							
		CDRF16[7:0]							
		—	—	—	—	—	—	—	—

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	GRCF16	GRCF16[23:16]							
		GRCF16[15:8]							
		GRCF16[7:0]							
		—	—	—	—	—	—	—	—
	TCRF17	CKSELF17[2:0]			MDF17[2:0]			EGSELF17[1:0]	
	TIERF17	—	—	—	—	OVECF17	OVEBF17	OVEAF17	ICEF17
	TSRF17	—	—	—	—	OVFCF17	OVFBF17	OVFAF17	ICFF17
	ECNTAF17	ECNTAF17[23:16]							
		ECNTAF17[15:8]							
		ECNTAF17[7:0]							
		—	—	—	—	—	—	—	—
	ECNTBF17	ECNTBF17[15:8]							
		ECNTBF17[7:0]							
	GRBF17	GRBF17[15:8]							
		GRBF17[7:0]							
	ECNTCF17	ECNTCF17[23:16]							
		ECNTCF17[15:8]							
		ECNTCF17[7:0]							
		—	—	—	—	—	—	—	—
	GRAF17	GRAF17[23:16]							
		GRAF17[15:8]							
		GRAF17[7:0]							
		—	—	—	—	—	—	—	—
	CDRF17	CDRF17[23:16]							
		CDRF17[15:8]							
		CDRF17[7:0]							
		—	—	—	—	—	—	—	—
	GRCF17	GRCF17[23:16]							
		GRCF17[15:8]							
		GRCF17[7:0]							
		—	—	—	—	—	—	—	—

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	TCRF18	CKSELF18[2:0]			MDF18[2:0]			EGSELF18[1:0]	
	TIERF18	—	—	—	—	OVECF18	OVEBF18	OVEAF18	ICEF18
	TSRF18	—	—	—	—	OVFCF18	OVFBF18	OVFAF18	ICFF18
	ECNTAF18	ECNTAF18[23:16]							
		ECNTAF18[15:8]							
		ECNTAF18[7:0]							
		—	—	—	—	—	—	—	—
	ECNTBF18	ECNTBF18[15:8]							
		ECNTBF18[7:0]							
	GRBF18	GRBF18[15:8]							
		GRBF18[7:0]							
	ECNTCF18	ECNTCF18[23:16]							
		ECNTCF18[15:8]							
		ECNTCF18[7:0]							
		—	—	—	—	—	—	—	—
	GRAF18	GRAF18[23:16]							
		GRAF18[15:8]							
		GRAF18[7:0]							
		—	—	—	—	—	—	—	—
	CDRF18	CDRF18[23:16]							
		CDRF18[15:8]							
		CDRF18[7:0]							
		—	—	—	—	—	—	—	—
	GRCF18	GRCF18[23:16]							
		GRCF18[15:8]							
		GRCF18[7:0]							
		—	—	—	—	—	—	—	—
	TCRF19	CKSELF19[2:0]			MDF19[2:0]			EGSELF19[1:0]	
	TIERF19	—	—	—	—	OVECF19	OVEBF19	OVEAF19	ICEF19
	TSRF19	—	—	—	—	OVFCF19	OVFBF19	OVFAF19	ICFF19

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	ECNTAF19	ECNTAF19[23:16]							
		ECNTAF19[15:8]							
		ECNTAF19[7:0]							
		—	—	—	—	—	—	—	—
	ECNTBF19	ECNTBF19[15:8]							
		ECNTBF19[7:0]							
	GRBF19	GRBF19[15:8]							
		GRBF19[7:0]							
	ECNTCF19	ECNTCF19[23:16]							
		ECNTCF19[15:8]							
		ECNTCF19[7:0]							
		—	—	—	—	—	—	—	—
	GRAF19	GRAF19[23:16]							
		GRAF19[15:8]							
		GRAF19[7:0]							
		—	—	—	—	—	—	—	—
	CDRF19	CDRF19[23:16]							
		CDRF19[15:8]							
		CDRF19[7:0]							
		—	—	—	—	—	—	—	—
	GRCF19	GRCF19[23:16]							
		GRCF19[15:8]							
		GRCF19[7:0]							
		—	—	—	—	—	—	—	—
	TCRF20	CKSELF20[2:0]			MDF20[2:0]			EGSELF20[1:0]	
	TIERF20	—	—	—	—	OVECF20	OVEBF20	OVEAF20	ICEF20
	TSRF20	—	—	—	—	OVFCF20	OVFBF20	OVFAF20	ICFF20
	ECNTAF20	ECNTAF20[23:16]							
		ECNTAF20[15:8]							
		ECNTAF20[7:0]							
		—	—	—	—	—	—	—	—

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	ECNTBF20	ECNTBF20[15:8]							
		ECNTBF20[7:0]							
	GRBF20	GRBF20[15:8]							
		GRBF20[7:0]							
	ECNTCF20	ECNTCF20[23:16]							
		ECNTCF20[15:8]							
		ECNTCF20[7:0]							
		—	—	—	—	—	—	—	—
	GRAF20	GRAF20[23:16]							
		GRAF20[15:8]							
		GRAF20[7:0]							
		—	—	—	—	—	—	—	—
	CDRF20	CDRF20[23:16]							
		CDRF20[15:8]							
		CDRF20[7:0]							
		—	—	—	—	—	—	—	—
	GRCF20	GRCF20[23:16]							
		GRCF20[15:8]							
		GRCF20[7:0]							
		—	—	—	—	—	—	—	—
	TCRF21	CKSELF21[2:0]			MDF21[2:0]			EGSELF21[1:0]	
	TIERF21	—	—	—	—	OVECF21	OVEBF21	OVEAF21	ICEF21
	TSRF21	—	—	—	—	OVFCF21	OVFBF21	OVFAF21	ICFF21
	ECNTAF21	ECNTAF21[23:16]							
		ECNTAF21[15:8]							
		ECNTAF21[7:0]							
		—	—	—	—	—	—	—	—
	ECNTBF21	ECNTBF21[15:8]							
		ECNTBF21[7:0]							
	GRBF21	GRBF21[15:8]							
		GRBF21[7:0]							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	ECNTCF21	ECNTCF21[23:16]							
		ECNTCF21[15:8]							
		ECNTCF21[7:0]							
		—	—	—	—	—	—	—	—
	GRAF21	GRAF21[23:16]							
		GRAF21[15:8]							
		GRAF21[7:0]							
		—	—	—	—	—	—	—	—
	CDRF21	CDRF21[23:16]							
		CDRF21[15:8]							
		CDRF21[7:0]							
		—	—	—	—	—	—	—	—
	GRCF21	GRCF21[23:16]							
		GRCF21[15:8]							
		GRCF21[7:0]							
		—	—	—	—	—	—	—	—
	TCRF22	CKSELF22[2:0]			MDF22[2:0]			EGSELF22[1:0]	
	TIERF22	—	—	—	—	OVECF22	OVEBF22	OVEAF22	ICEF22
	TSRF22	—	—	—	—	OVFCF22	OVFBF22	OVFAF22	ICFF22
	ECNTAF22	ECNTAF22[23:16]							
		ECNTAF22[15:8]							
		ECNTAF22[7:0]							
		—	—	—	—	—	—	—	—
	ECNTBF22	ECNTBF22[15:8]							
		ECNTBF22[7:0]							
	GRBF22	GRBF22[15:8]							
		GRBF22[7:0]							
	ECNTCF22	ECNTCF22[23:16]							
		ECNTCF22[15:8]							
		ECNTCF22[7:0]							
		—	—	—	—	—	—	—	—

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	GRAF22	GRAF22[23:16]							
		GRAF22[15:8]							
		GRAF22[7:0]							
		—	—	—	—	—	—	—	—
	CDRF22	CDRF22[23:16]							
		CDRF22[15:8]							
		CDRF22[7:0]							
		—	—	—	—	—	—	—	—
	GRCF22	GRCF22[23:16]							
		GRCF22[15:8]							
		GRCF22[7:0]							
		—	—	—	—	—	—	—	—
	TCRF23	CKSELF23[2:0]			MDF23[2:0]			EGSELF23[1:0]	
	TIERF23	—	—	—	—	OVECF23	OVEBF23	OVEAF23	ICEF23
	TSRF23	—	—	—	—	OVFCF23	OVFBF23	OVFAF23	ICFF23
	ECNTAF23	ECNTAF23[23:16]							
		ECNTAF23[15:8]							
		ECNTAF23[7:0]							
		—	—	—	—	—	—	—	—
	ECNTBF23	ECNTBF23[15:8]							
		ECNTBF23[7:0]							
	GRBF23	GRBF23[15:8]							
		GRBF23[7:0]							
	ECNTCF23	ECNTCF23[23:16]							
		ECNTCF23[15:8]							
		ECNTCF23[7:0]							
		—	—	—	—	—	—	—	—
	GRAF23	GRAF23[23:16]							
		GRAF23[15:8]							
		GRAF23[7:0]							
		—	—	—	—	—	—	—	—

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	CDRF23	CDRF23[23:16]							
		CDRF23[15:8]							
		CDRF23[7:0]							
		—	—	—	—	—	—	—	—
	GRCF23	GRCF23[23:16]							
		GRCF23[15:8]							
		GRCF23[7:0]							
		—	—	—	—	—	—	—	—
	TCRF24	CKSELF24[2:0]			MDF24[2:0]			EGSELF24[1:0]	
	TIERF24	—	—	—	—	OVECF24	OVEBF24	OVEAF24	ICEF24
	TSRF24	—	—	—	—	OVFCF24	OVFBF24	OVFAF24	ICFF24
	ECNTAF24	ECNTAF24[23:16]							
		ECNTAF24[15:8]							
		ECNTAF24[7:0]							
		—	—	—	—	—	—	—	—
	ECNTBF24	ECNTBF24[15:8]							
		ECNTBF24[7:0]							
	GRBF24	GRBF24[15:8]							
		GRBF24[7:0]							
	ECNTCF24	ECNTCF24[23:16]							
		ECNTCF24[15:8]							
		ECNTCF24[7:0]							
		—	—	—	—	—	—	—	—
	GRAF24	GRAF24[23:16]							
		GRAF24[15:8]							
		GRAF24[7:0]							
		—	—	—	—	—	—	—	—
	CDRF24	CDRF24[23:16]							
		CDRF24[15:8]							
		CDRF24[7:0]							
		—	—	—	—	—	—	—	—

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	GRCF24	GRCF24[23:16]							
		GRCF24[15:8]							
		GRCF24[7:0]							
		—	—	—	—	—	—	—	—
	TCRF25	CKSELF25[2:0]			MDF25[2:0]			EGSELF25[1:0]	
	TIERF25	—	—	—	—	OVECF25	OVEBF25	OVEAF25	ICEF25
	TSRF25	—	—	—	—	OVFCF25	OVFBF25	OVFAF25	ICFF25
	ECNTAF25	ECNTAF25[23:16]							
		ECNTAF25[15:8]							
		ECNTAF25[7:0]							
		—	—	—	—	—	—	—	—
	ECNTBF25	ECNTBF25[15:8]							
		ECNTBF25[7:0]							
	GRBF25	GRBF25[15:8]							
		GRBF25[7:0]							
	ECNTCF25	ECNTCF25[23:16]							
		ECNTCF25[15:8]							
		ECNTCF25[7:0]							
		—	—	—	—	—	—	—	—
	GRAF25	GRAF25[23:16]							
		GRAF25[15:8]							
		GRAF25[7:0]							
		—	—	—	—	—	—	—	—
	CDRF25	CDRF25[23:16]							
		CDRF25[15:8]							
		CDRF25[7:0]							
		—	—	—	—	—	—	—	—
	GRCF25	GRCF25[23:16]							
		GRCF25[15:8]							
		GRCF25[7:0]							
		—	—	—	—	—	—	—	—

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	TCRF26	CKSELF26[2:0]			MDF26[2:0]			EGSELF26[1:0]	
	TIERF26	—	—	—	—	OVECF26	OVEBF26	OVEAF26	ICEF26
	TSRF26	—	—	—	—	OVFCF26	OVFBF26	OVFAF26	ICFF26
	ECNTAF26	ECNTAF26[23:16]							
		ECNTAF26[15:8]							
		ECNTAF26[7:0]							
		—	—	—	—	—	—	—	—
	ECNTBF26	ECNTBF26[15:8]							
		ECNTBF26[7:0]							
	GRBF26	GRBF26[15:8]							
		GRBF26[7:0]							
	ECNTCF26	ECNTCF26[23:16]							
		ECNTCF26[15:8]							
		ECNTCF26[7:0]							
		—	—	—	—	—	—	—	—
	GRAF26	GRAF26[23:16]							
		GRAF26[15:8]							
		GRAF26[7:0]							
		—	—	—	—	—	—	—	—
	CDRF26	CDRF26[23:16]							
		CDRF26[15:8]							
		CDRF26[7:0]							
		—	—	—	—	—	—	—	—
	GRCF26	GRCF26[23:16]							
		GRCF26[15:8]							
		GRCF26[7:0]							
		—	—	—	—	—	—	—	—
	TCRF27	CKSELF27[2:0]			MDF27[2:0]			EGSELF27[1:0]	
	TIERF27	—	—	—	—	OVECF27	OVEBF27	OVEAF27	ICEF27
	TSRF27	—	—	—	—	OVFCF27	OVFBF27	OVFAF27	ICFF27

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	ECNTAF27	ECNTAF27[23:16]							
		ECNTAF27[15:8]							
		ECNTAF27[7:0]							
		—	—	—	—	—	—	—	—
	ECNTBF27	ECNTBF27[15:8]							
		ECNTBF27[7:0]							
	GRBF27	GRBF27[15:8]							
		GRBF27[7:0]							
	ECNTCF27	ECNTCF27[23:16]							
		ECNTCF27[15:8]							
		ECNTCF27[7:0]							
		—	—	—	—	—	—	—	—
	GRAF27	GRAF27[23:16]							
		GRAF27[15:8]							
		GRAF27[7:0]							
		—	—	—	—	—	—	—	—
	CDRF27	CDRF27[23:16]							
		CDRF27[15:8]							
		CDRF27[7:0]							
		—	—	—	—	—	—	—	—
	GRCF27	GRCF27[23:16]							
		GRCF27[15:8]							
		GRCF27[7:0]							
		—	—	—	—	—	—	—	—
	TSTRG	—	—	STRG5	STRG4	STRG3	STRG2	STRG1	STRG0
	TCRG0	—	CKSELG0[2:0]			—	—	CMPOEG0	CMEG0
	TSRG0	—	—	—	—	—	—	OVFG0	CMFG0
	TCNTG0	TCNTG0[15:8]							
		TCNTG0[7:0]							
	OCRG0	OCRG0[15:8]							
		OCRG0[7:0]							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	TCRG1	—	CKSELG1[2:0]			—	—	CMPOEG1	CMEG1
	TSRG1	—	—	—	—	—	—	OVFG1	CMFG1
	TCNTG1	TCNTG1[15:8]							
		TCNTG1[7:0]							
	OCRG1	OCRG1[15:8]							
		OCRG1[7:0]							
	TCRG2	—	CKSELG2[2:0]			—	—	CMPOEG2	CMEG2
	TSRG2	—	—	—	—	—	—	OVFG2	CMFG2
	TCNTG2	TCNTG2[15:8]							
		TCNTG2[7:0]							
	OCRG2	OCRG2[15:8]							
		OCRG2[7:0]							
	TCRG3	—	CKSELG3[2:0]			—	—	CMPOEG3	CMEG3
	TSRG3	—	—	—	—	—	—	OVFG3	CMFG3
	TCNTG3	TCNTG3[15:8]							
		TCNTG3[7:0]							
	OCRG3	OCRG3[15:8]							
		OCRG3[7:0]							
	TCRG4	—	CKSELG4[2:0]			—	—	CMPOEG4	CMEG4
	TSRG4	—	—	—	—	—	—	OVFG4	CMFG4
	TCNTG4	TCNTG4[15:8]							
		TCNTG4[7:0]							
	OCRG4	OCRG4[15:8]							
		OCRG4[7:0]							
	TCRG5	—	CKSELG5[2:0]			—	—	CMPOEG5	CMEG5
	TSRG5	—	—	—	—	—	—	OVFG5	CMFG5
	TCNTG5	TCNTG5[15:8]							
		TCNTG5[7:0]							
	OCRG5	OCRG5[15:8]							
		OCRG5[7:0]							
	TCRH	—	CKSELH[2:0]			—	—	—	CMEH
	TSRH	—	—	—	—	—	OVF2H	OVF1H	CMFH

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	TCNT1H	TCNT1H[15:8]							
		TCNT1H[7:0]							
	OCR1H	OCR1H[15:8]							
		OCR1H[7:0]							
	TCNT2H	TCNT2H[31:24]							
		TCNT2H[23:16]							
		TCNT2H[15:8]							
		TCNT2H[7:0]							
	TSTRJ	—	—	—	—	—	—	STRJ1	STRJ0
	TCRJ0	—	CKSELJ0[2:0]			—	NCEJ0	IOJ0[1:0]	
	FCRJ0	FIFOENJ0	—	FVCRENJ0	FRSTJ0	—	—	FDFTRGJ0[1:0]	
	TSRJ0	—	—	—	FVLDFJ0	CMFJ0	OVFJ0	FDOVFJ0	FDFFJ0
	TIERJ0	—	—	—	—	—	OVEJ0	FDOVEJ0	FDFEJ0
	FDNRJ0	—	—	—	—	FDNJ0[3:0]			
	NCNTJ0	NCCNTJ0[7:0]							
	NCRJ0	NCTJ0[7:0]							
	TCNTJ0	TCNTJ0[15:8]							
		TCNTJ0[7:0]							
	OCRJ0	OCRJ0[15:8]							
		OCRJ0[7:0]							
	FIFOJ0	FIFODJ0[15:8]							
		FIFODJ0[7:0]							
	TCRJ1	—	CKSELJ1[2:0]			—	NCEJ1	IOJ1[1:0]	
	FCRJ1	FIFOENJ1	—	FVCRENJ1	FRSTJ1	—	—	FDFTRGJ1[1:0]	
	TSRJ1	—	—	—	FVLDFJ1	CMFJ1	OVFJ1	FDOVFJ1	FDFFJ1
	TIERJ1	—	—	—	—	—	OVEJ1	FDOVEJ1	FDFEJ1
	FDNRJ1	—	—	—	—	FDNJ1[3:0]			
	NCNTJ1	NCCNTJ1[7:0]							
	NCRJ1	NCTJ1[7:0]							
	TCNTJ1	TCNTJ1[15:8]							
		TCNTJ1[7:0]							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ATU-III	OCRJ1	OCRJ1[15:8]							
		OCRJ1[7:0]							
	FIFOJ1	FIFODJ1[15:8]							
		FIFODJ1[7:0]							
WDT	WTCR	TCRKEY[7:0]							
		—	WT/$\overline{\text{IT}}$	TME	—	—	CKS[2:0]		
	WTCNT	TCNTKEY[7:0]							
		TCNT[7:0]							
	WTSR	TSRKEY[7:0]							
		WOVF	—	—	—	IOVF	—	—	—
	WRCR	RCRKEY[7:0]							
		RSTE	—	—	—	—	—	—	—
CMT	CMSTR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	STR1	STR0
	CMCR_0	—	CMIE	—	—	—	—	CKS[1:0]	
	CMSR_0	—	—	—	—	—	—	—	CMF
	CMCNT_0								
	CMCOR_0								
	CMCR_1	—	CMIE	—	—	—	—	CKS[1:0]	
	CMSR_1	—	—	—	—	—	—	—	CMF
	CMCNT_1								
	CMCOR_1								
SCI	SCSMR1A	C/$\overline{\text{A}}$	CHR	PE	O/$\overline{\text{E}}$	STOP	—	CKS[1:0]	
	SCBRR1A								
	SCSCR1A	TIE	RIE	TE	RE	—	TEIE	CKE1	—
	SCTDR1A								
	SCSSR1A	TDRE	RDRF	ORER	FER	PER	TEND	—	—
	SCRDR1A								

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
SCI	SCSMR1B	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	—	CKS[1:0]	
	SCBRR1B								
	SCSCR1B	TIE	RIE	TE	RE	—	TEIE	CKE1	—
	SCTDR1B								
	SCSSR1B	TDRE	RDRF	ORER	FER	PER	TEND	—	—
	SCRDR1B								
	SCSMR1C	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	—	CKS[1:0]	
	SCBRR1C								
	SCSCR1C	TIE	RIE	TE	RE	—	TEIE	CKE1	—
	SCTDR1C								
	SCSSR1C	TDRE	RDRF	ORER	FER	PER	TEND	—	—
	SCRDR1C								
	SCSMR1D	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	—	CKS[1:0]	
	SCBRR1D								
	SCSCR1D	TIE	RIE	TE	RE	—	TEIE	CKE1	—
	SCTDR1D								
	SCSSR1D	TDRE	RDRF	ORER	FER	PER	TEND	—	—
	SCRDR1D								
	SCSMR1E	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	—	CKS[1:0]	
	SCBRR1E								
	SCSCR1E	TIE	RIE	TE	RE	—	TEIE	CKE1	—
	SCTDR1E								
	SCSSR1E	TDRE	RDRF	ORER	FER	PER	TEND	—	—
	SCRDR1E								
RSPI	SPCRA	SPRIE	SPE	SPTIE	SPEIE	MSTR	MODFEN	—	—
	SSLPA	SSL7P	SSL6P	SSL5P	SSL4P	SSL3P	SSL2P	SSL1P	SSL0P
	SPPCRA	—	—	MOIFE	MOIFV	—	SPOM	—	SPLP
	SPSRA	SPRF	—	SPTEF	—	—	MODF	—	OVRF
	SPDRA	SPD15	SPD14	SPD13	SPD12	SPD11	SPD10	SPD9	SPD8
		SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0
	SPSCRA	—	—	—	—	—	SPSLN[2:0]		
	SPSSRA	—	SPECM[2:0]			—	SPCP[2:0]		

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
RSPI	SPBRA	SPR7	SPR6	SPR5	SPR4	SPR3	SPR2	SPR1	SPR0
	SPCKDA	—	—	—	—	—	SCKDL[2:0]		
	SSLNDA	—	—	—	—	—	SLNDL[2:0]		
	SPNDA	—	—	—	—	—	SPNDL[2:0]		
	SPCMDA0	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]			
		SSLKP	SSLA[2:0]			BRDV[1:0]		CPOL	CPHA
	SPCMDA1	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]			
		SSLKP	SSLA[2:0]			BRDV[1:0]		CPOL	CPHA
	SPCMDA2	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]			
		SSLKP	SSLA[2:0]			BRDV[1:0]		CPOL	CPHA
	SPCMDA3	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]			
		SSLKP	SSLA[2:0]			BRDV[1:0]		CPOL	CPHA
	SPCMDA4	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]			
		SSLKP	SSLA[2:0]			BRDV[1:0]		CPOL	CPHA
	SPCMDA5	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]			
		SSLKP	SSLA[2:0]			BRDV[1:0]		CPOL	CPHA
	SPCMDA6	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]			
		SSLKP	SSLA[2:0]			BRDV[1:0]		CPOL	CPHA
	SPCMDA7	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]			
		SSLKP	SSLA[2:0]			BRDV[1:0]		CPOL	CPHA
	SPCRB	SPRIE	SPE	SPTIE	SPEIE	MSTR	MODFEN	—	—
	SSLPB	SSL7P	SSL6P	SSL5P	SSL4P	SSL3P	SSL2P	SSL1P	SSL0P
	SPPCRB	—	—	MOIFE	MOIFV	—	SPOM	—	SPLP
	SPSRB	SPRF	—	SPTEF	—	—	MODF	—	OVRF
	SPDRB	SPD15	SPD14	SPD13	SPD12	SPD11	SPD10	SPD9	SPD8
		SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0
	SPSCRB	—	—	—	—	—	SPSLN[2:0]		
	SPSSRB	—	SPECM[2:0]			—	SPCP[2:0]		
	SPBRB	SPR7	SPR6	SPR5	SPR4	SPR3	SPR2	SPR1	SPR0
	SPCKDB	—	—	—	—	—	SCKDL[2:0]		
	SSLNDB	—	—	—	—	—	SLNDL[2:0]		
	SPNDB	—	—	—	—	—	SPNDL[2:0]		

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
RSPI	SPCMDB0	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]			
		SSLKP	SSLA[2:0]			BRDV[1:0]		CPOL	CPHA
	SPCMDB1	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]			
		SSLKP	SSLA[2:0]			BRDV[1:0]		CPOL	CPHA
	SPCMDB2	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]			
		SSLKP	SSLA[2:0]			BRDV[1:0]		CPOL	CPHA
	SPCMDB3	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]			
		SSLKP	SSLA[2:0]			BRDV[1:0]		CPOL	CPHA
	SPCMDB4	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]			
		SSLKP	SSLA[2:0]			BRDV[1:0]		CPOL	CPHA
	SPCMDB5	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]			
		SSLKP	SSLA[2:0]			BRDV[1:0]		CPOL	CPHA
	SPCMDB6	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]			
		SSLKP	SSLA[2:0]			BRDV[1:0]		CPOL	CPHA
	SPCMDB7	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]			
		SSLKP	SSLA[2:0]			BRDV[1:0]		CPOL	CPHA
	SPCRC	SPRIE	SPE	SPTIE	SPEIE	MSTR	MODFEN	—	—
	SSLPC	SSL7P	SSL6P	SSL5P	SSL4P	SSL3P	SSL2P	SSL1P	SSL0P
	SPPCRC	—	—	MOIFE	MOIFV	—	SPOM	—	SPLP
	SPSRC	SPRF	—	SPTEF	—	—	MODF	—	OVRF
	SPDRC	SPD15	SPD14	SPD13	SPD12	SPD11	SPD10	SPD9	SPD8
		SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0
	SPSCRC	—	—	—	—	—	SPSLN[2:0]		
	SPSSRC	—	SPECM[2:0]			—	SPCP[2:0]		
	SPBRC	SPR7	SPR6	SPR5	SPR4	SPR3	SPR2	SPR1	SPR0
	SPCKDC	—	—	—	—	—	SCKDL[2:0]		
	SSLNDC	—	—	—	—	—	SLNDL[2:0]		
	SPNDC	—	—	—	—	—	SPNDL[2:0]		
	SPCMDC0	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]			
		SSLKP	SSLA[2:0]			BRDV[1:0]		CPOL	CPHA
	SPCMDC1	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]			
		SSLKP	SSLA[2:0]			BRDV[1:0]		CPOL	CPHA

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
RSPI	SPCMDC2	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]			
		SSLKP	SSLA[2:0]			BRDV[1:0]		CPOL	CPHA
	SPCMDC3	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]			
		SSLKP	SSLA[2:0]			BRDV[1:0]		CPOL	CPHA
	SPCMDC4	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]			
		SSLKP	SSLA[2:0]			BRDV[1:0]		CPOL	CPHA
	SPCMDC5	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]			
		SSLKP	SSLA[2:0]			BRDV[1:0]		CPOL	CPHA
	SPCMDC6	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]			
		SSLKP	SSLA[2:0]			BRDV[1:0]		CPOL	CPHA
	SPCMDC7	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]			
		SSLKP	SSLA[2:0]			BRDV[1:0]		CPOL	CPHA
RCAN-TL1 (RCAN_A)	MCR	MCR15	MCR14	—	—	—	TST[2:0]		
		MCR7	MCR6	MCR5	—	—	MCR2	MCR1	MCR0
	GSR	—	—	—	—	—	—	—	—
		—	—	GSR5	GSR4	GSR3	GSR2	GSR1	GSR0
	BCR1	TSG1[3:0]				—	TSG2[2:0]		
		—	—	SJW[1:0]		—	—	—	BSP
	BCR0	—	—	—	—	—	—	—	—
		BRP[7:0]							
	IRR	IRR15	IRR14	IRR13	IRR12	IRR11	IRR10	IRR9	IRR8
		IRR7	IRR6	IRR5	IRR4	IRR3	IRR2	IRR1	IRR0
	IMR	IMR15	IMR14	IMR13	IMR12	IMR11	IMR10	IMR9	IMR8
		IMR7	IMR6	IMR5	IMR4	IMR3	IMR2	IMR1	IMR0
	TEC/REC	TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0
		REC7	REC6	REC5	REC4	REC3	REC2	REC1	REC0
	TXPR1	TXPR1[15:8]							
		TXPR1[7:0]							
	TXPR0	TXPR0[15:8]							
		TXPR0[7:1]							—
	TXCR1	TXCR1[15:8]							
		TXCR1[7:0]							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
RCAN-TL1 (RCAN_A)	TXCR0	TXCR0[15:8]							
		TXCR0[7:1]							—
	TXACK1	TXACK1[15:8]							
		TXACK1[7:0]							
	TXACK0	TXACK0[15:8]							
		TXACK0[7:1]							—
	ABACK1	ABACK1[15:8]							
		ABACK1[7:0]							
	ABACK0	ABACK0[15:8]							
		ABACK0[7:1]							—
	RXPR1	RXPR1[15:8]							
		RXPR1[7:0]							
	RXPR0	RXPR0[15:8]							
		RXPR0[7:0]							
	RFPR1	RFPR1[15:8]							
		RFPR1[7:0]							
	RFPR0	RFPR0[15:8]							
		RFPR0[7:0]							
	MBIMR1	MBIMR1[15:8]							
		MBIMR1[7:0]							
	MBIMR0	MBIMR0[15:8]							
		MBIMR0[7:0]							
	UMSR1	UMSR1[15:8]							
		UMSR1[7:0]							
	UMSR0	UMSR0[15:8]							
		UMSR0[7:0]							
	TTCR0	TCR15	TCR14	TCR13	TCR12	TCR11	TCR10	—	—
		—	TCR6	TPSC5	TPSC4	TPSC3	TPSC2	TPSC1	TPSC0
	CMAX_TEW	—	—	—	—	—	CMAX[2:0]		
		—	—	—	—	TEW[3:0]			
	RFTROFF	RFTROFF[7:0]							
		—	—	—	—	—	—	—	—

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
RCAN-TL1 (RCAN_A)	TSR	—	—	—	—	—	—	—	—
		—	—	—	TSR4	TSR3	TSR2	TSR1	TSR0
	CCR	—	—	—	—	—	—	—	—
		—	—	CCR[5:0]					
	TCNTR	TCNTR[15:8]							
		TCNTR[7:0]							
	CYCTR	CYCTR[15:8]							
		CYCTR[7:0]							
	RFMK	RFMK[15:8]							
		RFMK[7:0]							
	TCMR0	TCMR0[15:8]							
		TCMR0[7:0]							
	TCMR1	TCMR1[15:8]							
		TCMR1[7:0]							
	TCMR2	TCMR2[15:8]							
		TCMR2[7:0]							
	TTTSEL	—	TTTSEL[14:8]						
		—	—	—	—	—	—	—	—
	MB[0]. CONTROL0_H	IDE	RTR	—	STDID[10:6]				
		STDID[5:0]						EXTID[17:16]	
	MB[0]. CONTROL0_L	EXTID[15:8]							
		EXTID[7:0]							
	MB[0]. LAFM0	IDE_LAFM	—	—	STDID_LAFM[10:6]				
		STDID_LAFM[5:0]						EXTID_LAFM[17:16]	
	MB[0]. LAFM1	EXTID_LAFM[15:8]							
		EXTID_LAFM[7:0]							
	MB[0]. DATA_01	MSG_DATA_0							
		MSG_DATA_1							
	MB[0]. DATA_23	MSG_DATA_2							
		MSG_DATA_3							
	MB[0]. DATA_45	MSG_DATA_4							
		MSG_DATA_5							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
RCAN-TL1 (RCAN_A)	MB[0]. DATA_67	MSG_DATA_6							
		MSG_DATA_7							
	MB[0]. CONTROL1	—	—	NMC	—	—	MBC[2:0]		
		—	—	—	—	DLC[3:0]			
	MB[0]. TIMESTAMP	TimeStamp[15:8]							
		TimeStamp[7:0]							
	MB[1]. CONTROL0_H	IDE	RTR	—	STDID[10:6]				
		STDID[5:0]						EXTID[17:16]	
	MB[1]. CONTROL0_L	EXTID[15:8]							
		EXTID[7:0]							
	MB[1]. LAFM0	IDE_LAFM	—	—	STDID_LAFM[10:6]				
		STDID_LAFM[5:0]						EXTID_LAFM[17:16]	
	MB[1]. LAFM1	EXTID_LAFM[15:8]							
		EXTID_LAFM[7:0]							
	MB[1]. DATA_01	MSG_DATA_0							
		MSG_DATA_1							
	MB[1]. DATA_23	MSG_DATA_2							
		MSG_DATA_3							
	MB[1]. DATA_45	MSG_DATA_4							
		MSG_DATA_5							
	MB[1]. DATA_67	MSG_DATA_6							
		MSG_DATA_7							
	MB[1]. CONTROL1	—	—	NMC	ATX	DART	MBC[2:0]		
		—	—	—	—	DLC[3:0]			
	MB[1]. TIMESTAMP	TimeStamp[15:8]							
		TimeStamp[7:0]							
	MB[2]	Configured in the same order as in MB[1]							
	:	:							
	MB[15]	Configured in the same order as in MB[1]							
	MB[16]. CONTROL0_H	IDE	RTR	—	STDID[10:6]				
		STDID[5:0]						EXTID[17:16]	

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
RCAN-TL1 (RCAN_A)	MB[16]. CONTROL0_L	EXTID[15:8]							
		EXTID[7:0]							
	MB[16]. LAFM0	IDE_LAFM	—	—	STDID_LAFM[10:6]				
		STDID_LAFM[5:0]						EXTID_LAFM[17:16]	
	MB[16]. LAFM1	EXTID_LAFM[15:8]							
		EXTID_LAFM[7:0]							
	MB[16]. DATA_01	MSG_DATA_0							
		MSG_DATA_1							
	MB[16]. DATA_23	MSG_DATA_2							
		MSG_DATA_3							
	MB[16]. DATA_45	MSG_DATA_4							
		MSG_DATA_5							
	MB[16]. DATA_67	MSG_DATA_6							
		MSG_DATA_7							
	MB[16]. CONTROL1	—	—	NMC	ATX	DART	MBC[2:0]		
		—	—	—	—	DLC[3:0]			
	MB[17]	Configured in the same order as in MB[16]							
	:	:							
	MB[23]	Configured in the same order as in MB[16]							
	MB[24]. CONTROL0_H	IDE	RTR	—	STDID[10:6]				
		STDID[5:0]						EXTID[17:16]	
	MB[24]. CONTROL0_L	EXTID[15:8]							
		EXTID[7:0]							
	MB[24]. LAFM0	IDE_LAFM	—	—	STDID_LAFM[10:6]				
		STDID_LAFM[5:0]						EXTID_LAFM[17:16]	
	MB[24]. LAFM1	EXTID_LAFM[15:8]							
		EXTID_LAFM[7:0]							
	MB[24]. DATA_01	MSG_DATA_0							
		MSG_DATA_1							
	MB[24]. DATA_23	MSG_DATA_2							
		MSG_DATA_3							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
RCAN-TL1 (RCAN_A)	MB[24]. DATA_45	MSG_DATA_4							
		MSG_DATA_5							
	MB[24]. DATA_67	MSG_DATA_6							
		MSG_DATA_7							
	MB[24]. CONTROL1	—	—	NMC	ATX	DART	MBC[2:0]		
		—	—	—	—	DLC[3:0]			
	MB[24]. TTT	TTT							
		TTT							
	MB[24]. TTCONTROL	TTW[1:0]		Offset					
		—	—	—	—	—	Rep_Factor		
	MB[25]	Configured in the same order as in MB[24]							
	:	:							
	MB[29]	Configured in the same order as in MB[24]							
	MB[30]. CONTROL0_H	IDE	RTR	—	STDID[10:6]				
		STDID[5:0]						EXTID[17:16]	
	MB[30]. CONTROL0_L	EXTID[15:8]							
		EXTID[7:0]							
	MB[30]. LAFM0	IDE_LAFM	—	—	STDID_LAFM[10:6]				
		STDID_LAFM[5:0]						EXTID_LAFM[17:16]	
	MB[30]. LAFM1	EXTID_LAFM[15:8]							
		EXTID_LAFM[7:0]							
	MB[30]. DATA_01	MSG_DATA_0							
		MSG_DATA_1							
	MB[30]. DATA_23	MSG_DATA_2							
		MSG_DATA_3							
	MB[30]. DATA_45	MSG_DATA_4							
		MSG_DATA_5							
	MB[30]. DATA_67	MSG_DATA_6							
		MSG_DATA_7							
	MB[30]. CONTROL1	—	—	NMC	ATX	DART	MBC[2:0]		
		—	—	—	—	DLC[3:0]			

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
RCAN- TL1 (RCAN_A)	MB[30]. TIMESTAMP	TimeStamp[15:8]							
		TimeStamp[7:0]							
	MB[30]. TTT	TTT							
		TTT							
	MB[31]. CONTROL0_H	IDE	RTR	—	STDID[10:6]				
		STDID[5:0]						EXTID[17:16]	
	MB[31]. CONTROL0_L	EXTID[15:8]							
		EXTID[7:0]							
	MB[31]. LAFM0	IDE_LAFM	—	—	STDID_LAFM[10:6]				
		STDID_LAFM[5:0]						EXTID_LAFM[17:16]	
	MB[31]. LAFM1	EXTID_LAFM[15:8]							
		EXTID_LAFM[7:0]							
	MB[31]. DATA_01	MSG_DATA_0							
		MSG_DATA_1							
	MB[31]. DATA_23	MSG_DATA_2							
		MSG_DATA_3							
	MB[31]. DATA_45	MSG_DATA_4							
		MSG_DATA_5							
	MB[31]. DATA_67	MSG_DATA_6							
		MSG_DATA_7							
	MB[31]. CONTROL1	—	—	NMC	ATX	DART	MBC[2:0]		
		—	—	—	—	DLC[3:0]			
	MB[31]. TIMESTAMP	TimeStamp[15:8]							
		TimeStamp[7:0]							
	MBESR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	MBEF
	MBECR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	MBIM

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
RCAN-TL1 (RCAN_B)	MCR	MCR15	MCR14	—	—	—	TST[2:0]		
		MCR7	MCR6	MCR5	—	—	MCR2	MCR1	MCR0
	GSR	—	—	—	—	—	—	—	—
		—	—	GSR5	GSR4	GSR3	GSR2	GSR1	GSR0
	BCR1	TSG1[3:0]				—	TSG2[2:0]		
		—	—	SJW[1:0]		—	—	—	BSP
	BCR0	—	—	—	—	—	—	—	—
		BRP[7:0]							
	IRR	IRR15	IRR14	IRR13	IRR12	IRR11	IRR10	IRR9	IRR8
		IRR7	IRR6	IRR5	IRR4	IRR3	IRR2	IRR1	IRR0
	IMR	IMR15	IMR14	IMR13	IMR12	IMR11	IMR10	IMR9	IMR8
		IMR7	IMR6	IMR5	IMR4	IMR3	IMR2	IMR1	IMR0
	TEC/REC	TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0
		REC7	REC6	REC5	REC4	REC3	REC2	REC1	REC0
	TXPR1	TXPR1[15:8]							
		TXPR1[7:0]							
	TXPR0	TXPR0[15:8]							
		TXPR0[7:1]							—
	TXCR1	TXCR1[15:8]							
		TXCR1[7:0]							
	TXCR0	TXCR0[15:8]							
		TXCR0[7:1]							—
	TXACK1	TXACK1[15:8]							
		TXACK1[7:0]							
	TXACK0	TXACK0[15:8]							
		TXACK0[7:1]							—
	ABACK1	ABACK1[15:8]							
		ABACK1[7:0]							
	ABACK0	ABACK0[15:8]							
		ABACK0[7:1]							—
	RXPR1	RXPR1[15:8]							
		RXPR1[7:0]							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
RCAN-TL1 (RCAN_B)	RXPR0	RXPR0[15:8]							
		RXPR0[7:0]							
	RFPR1	RFPR1[15:8]							
		RFPR1[7:0]							
	RFPR0	RFPR0[15:8]							
		RFPR0[7:0]							
	MBIMR1	MBIMR1[15:8]							
		MBIMR1[7:0]							
	MBIMR0	MBIMR0[15:8]							
		MBIMR0[7:0]							
	UMSR1	UMSR1[15:8]							
		UMSR1[7:0]							
	UMSR0	UMSR0[15:8]							
		UMSR0[7:0]							
	TTCR0	TCR15	TCR14	TCR13	TCR12	TCR11	TCR10	—	—
		—	TCR6	TPSC5	TPSC4	TPSC3	TPSC2	TPSC1	TPSC0
	CMAX_TEW	—	—	—	—	—	CMAX[2:0]		
		—	—	—	—	TEW[3:0]			
	RFTROFF	RFTROFF[7:0]							
		—	—	—	—	—	—	—	—
	TSR	—	—	—	—	—	—	—	—
		—	—	—	TSR4	TSR3	TSR2	TSR1	TSR0
	CCR	—	—	—	—	—	—	—	—
		—	—	CCR[5:0]					
	TCNTR	TCNTR[15:8]							
		TCNTR[7:0]							
	CYCTR	CYCTR[15:8]							
		CYCTR[7:0]							
	RFMK	RFMK[15:8]							
		RFMK[7:0]							
	TCMR0	TCMR0[15:8]							
		TCMR0[7:0]							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
RCAN-TL1 (RCAN_B)	TCMR1	TCMR1[15:8]							
		TCMR1[7:0]							
	TCMR2	TCMR2[15:8]							
		TCMR2[7:0]							
	TTTSEL	—	TTTSEL[14:8]						
		—	—	—	—	—	—	—	—
	MB[0]	Configured in the same order as in MB[0] of the RCAN_A.							
	MB[1]	Configured in the same order as in MB[1] of the RCAN_A.							
	MB[2]	Configured in the same order as in MB[2] of the RCAN_A.							
	:	:							
	MB[29]	Configured in the same order as in MB[29] of the RCAN_A.							
	MB[30]	Configured in the same order as in MB[30] of the RCAN_A.							
	MB[31]	Configured in the same order as in MB[31] of the RCAN_A.							
	MBESR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	MBEF
	MBECR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	MBIM
RCAN-TL1 (RCAN_C)	MCR	MCR15	MCR14	—	—	—	TST[2:0]		
		MCR7	MCR6	MCR5	—	—	MCR2	MCR1	MCR0
	GSR	—	—	—	—	—	—	—	—
		—	—	GSR5	GSR4	GSR3	GSR2	GSR1	GSR0
	BCR1	TSG1[3:0]				—	TSG2[2:0]		
		—	—	SJW[1:0]		—	—	—	BSP
	BCR0	—	—	—	—	—	—	—	—
		BRP[7:0]							
	IRR	IRR15	IRR14	IRR13	IRR12	IRR11	IRR10	IRR9	IRR8
		IRR7	IRR6	IRR5	IRR4	IRR3	IRR2	IRR1	IRR0
	IMR	IMR15	IMR14	IMR13	IMR12	IMR11	IMR10	IMR9	IMR8
		IMR7	IMR6	IMR5	IMR4	IMR3	IMR2	IMR1	IMR0
	TEC/REC	TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0
		REC7	REC6	REC5	REC4	REC3	REC2	REC1	REC0

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
RCAN-TL1 (RCAN_C)	TXPR1	TXPR1[15:8]							
		TXPR1[7:0]							
	TXPR0	TXPR0[15:8]							
		TXPR0[7:1]							—
	TXCR1	TXCR1[15:8]							
		TXCR1[7:0]							
	TXCR0	TXCR0[15:8]							
		TXCR0[7:1]							—
	TXACK1	TXACK1[15:8]							
		TXACK1[7:0]							
	TXACK0	TXACK0[15:8]							
		TXACK0[7:1]							—
	ABACK1	ABACK1[15:8]							
		ABACK1[7:0]							
	ABACK0	ABACK0[15:8]							
		ABACK0[7:1]							—
	RXPR1	RXPR1[15:8]							
		RXPR1[7:0]							
	RXPR0	RXPR0[15:8]							
		RXPR0[7:0]							
	RFPR1	RFPR1[15:8]							
		RFPR1[7:0]							
	RFPR0	RFPR0[15:8]							
		RFPR0[7:0]							
	MBIMR1	MBIMR1[15:8]							
		MBIMR1[7:0]							
	MBIMR0	MBIMR0[15:8]							
		MBIMR0[7:0]							
	UMSR1	UMSR1[15:8]							
		UMSR1[7:0]							
	UMSR0	UMSR0[15:8]							
		UMSR0[7:0]							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
RCAN-TL1 (RCAN_C)	TTCR0	TCR15	TCR14	TCR13	TCR12	TCR11	TCR10	—	—
		—	TCR6	TPSC5	TPSC4	TPSC3	TPSC2	TPSC1	TPSC0
	CMAX_TEW	—	—	—	—	—	CMAX[2:0]		
		—	—	—	—	TEW[3:0]			
	RFTROFF	RFTROFF[7:0]							
		—	—	—	—	—	—	—	—
	TSR	—	—	—	—	—	—	—	—
		—	—	—	TSR4	TSR3	TSR2	TSR1	TSR0
	CCR	—	—	—	—	—	—	—	—
		—	—	CCR[5:0]					
	TCNTR	TCNTR[15:8]							
		TCNTR[7:0]							
	CYCTR	CYCTR[15:8]							
		CYCTR[7:0]							
	RFMK	RFMK[15:8]							
		RFMK[7:0]							
	TCMR0	TCMR0[15:8]							
		TCMR0[7:0]							
	TCMR1	TCMR1[15:8]							
		TCMR1[7:0]							
	TCMR2	TCMR2[15:8]							
		TCMR2[7:0]							
	TTTSEL	—	TTTSEL[14:8]						
		—	—	—	—	—	—	—	—
	MB[0]	Configured in the same order as in MB[0] of the RCAN_A.							
	MB[1]	Configured in the same order as in MB[1] of the RCAN_A.							
	MB[2]	Configured in the same order as in MB[2] of the RCAN_A.							
	:	:							
	MB[29]	Configured in the same order as in MB[29] of the RCAN_A.							
	MB[30]	Configured in the same order as in MB[30] of the RCAN_A.							
	MB[31]	Configured in the same order as in MB[31] of the RCAN_A.							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
RCAN-TL1 (RCAN_C)	MBESR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	MBEF
	MBECR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	MBIM
RCAN-TL1 (RCAN_D)	MCR	MCR15	MCR14	—	—	—	TST[2:0]		
		MCR7	MCR6	MCR5	—	—	MCR2	MCR1	MCR0
	GSR	—	—	—	—	—	—	—	—
		—	—	GSR5	GSR4	GSR3	GSR2	GSR1	GSR0
	BCR1	TSG1[3:0]				—	TSG2[2:0]		
		—	—	SJW[1:0]		—	—	—	BSP
	BCR0	—	—	—	—	—	—	—	—
		BRP[7:0]							
	IRR	IRR15	IRR14	IRR13	IRR12	IRR11	IRR10	IRR9	IRR8
		IRR7	IRR6	IRR5	IRR4	IRR3	IRR2	IRR1	IRR0
	IMR	IMR15	IMR14	IMR13	IMR12	IMR11	IMR10	IMR9	IMR8
		IMR7	IMR6	IMR5	IMR4	IMR3	IMR2	IMR1	IMR0
	TEC/REC	TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0
		REC7	REC6	REC5	REC4	REC3	REC2	REC1	REC0
	TXPR1	TXPR1[15:8]							
		TXPR1[7:0]							
	TXPR0	TXPR0[15:8]							
		TXPR0[7:1]							—
	TXCR1	TXCR1[15:8]							
		TXCR1[7:0]							
	TXCR0	TXCR0[15:8]							
		TXCR0[7:1]							—
	TXACK1	TXACK1[15:8]							
		TXACK1[7:0]							
	TXACK0	TXACK0[15:8]							
		TXACK0[7:1]							—
	ABACK1	ABACK1[15:8]							
		ABACK1[7:0]							

Module	**Register Name**	**31/23/15/7**	**30/22/14/6**	**29/21/13/5**	**28/20/12/4**	**27/19/11/3**	**26/18/10/2**	**25/17/9/1**	**24/16/8/0**
RCAN- TL1 (RCAN_D)	ABACK0	ABACK0[15:8]							
		ABACK0[7:1]							—
	RXPR1	RXPR1[15:8]							
		RXPR1[7:0]							
	RXPR0	RXPR0[15:8]							
		RXPR0[7:0]							
	RFPR1	RFPR1[15:8]							
		RFPR1[7:0]							
	RFPR0	RFPR0[15:8]							
		RFPR0[7:0]							
	MBIMR1	MBIMR1[15:8]							
		MBIMR1[7:0]							
	MBIMR0	MBIMR0[15:8]							
		MBIMR0[7:0]							
	UMSR1	UMSR1[15:8]							
		UMSR1[7:0]							
	UMSR0	UMSR0[15:8]							
		UMSR0[7:0]							
	TTCR0	TCR15	TCR14	TCR13	TCR12	TCR11	TCR10	—	—
		—	TCR6	TPSC5	TPSC4	TPSC3	TPSC2	TPSC1	TPSC0
	CMAX_TEW	—	—	—	—	—	CMAX[2:0]		
		—	—	—	—	TEW[3:0]			
	RFTROFF	RFTROFF[7:0]							
		—	—	—	—	—	—	—	—
	TSR	—	—	—	—	—	—	—	—
		—	—	—	TSR4	TSR3	TSR2	TSR1	TSR0
	CCR	—	—	—	—	—	—	—	—
		—	—	CCR[5:0]					
	TCNTR	TCNTR[15:8]							
		TCNTR[7:0]							
	CYCTR	CYCTR[15:8]							
		CYCTR[7:0]							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
RCAN-TL1 (RCAN_D)	RFMK	RFMK[15:8]							
		RFMK[7:0]							
	TCMR0	TCMR0[15:8]							
		TCMR0[7:0]							
	TCMR1	TCMR1[15:8]							
		TCMR1[7:0]							
	TCMR2	TCMR2[15:8]							
		TCMR2[7:0]							
	TTTSEL	—	TTTSEL[14:8]						
		—	—	—	—	—	—	—	—
	MB[0]	Configured in the same order as in MB[0] of the RCAN_A.							
	MB[1]	Configured in the same order as in MB[1] of the RCAN_A.							
	MB[2]	Configured in the same order as in MB[2] of the RCAN_A.							
	:	:							
	MB[29]	Configured in the same order as in MB[29] of the RCAN_A.							
	MB[30]	Configured in the same order as in MB[30] of the RCAN_A.							
	MB[31]	Configured in the same order as in MB[31] of the RCAN_A.							
	MBESR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	MBEF
	MBECR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	MBIM
FlexRay	FXROC	FOPC	—	—	—	—	FBSEN	—	FOPEN
	FXROS	—	—	—	—	—	—	—	FRSTAT
	FXRTISR	—	—	—	—	—	—	FT1IS	FT0IS
	FXRTIER	—	—	—	—	—	—	FT1IEN	FT0IEN
	FRLCK	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		FRLCK							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
FlexRay	FREIR	—	—	—	—	—	TABB	LTVB	EDB
		—	—	—	—	—	TABA	LTVA	EDA
		—	—	—	—	MHF	IOBA	IIBA	EFA
		RFO	PERR	CCL	CCF	SFO	SFBM	CNA	PEMC
	FRSIR	—	—	—	—	—	—	MTSB	WUPB
		—	—	—	—	—	—	MTSA	WUPA
		SDS	MBSI	SUCS	SWE	TOBC	TIBC	TI1	TI0
		NMVC	RFCL	RFNE	RXI	TXI	CYCS	CAS	WST
	FREILS	—	—	—	—	—	TABBL	LTVBL	EDBL
		—	—	—	—	—	TABAL	LTVAL	EDAL
		—	—	—	—	MHFL	IOBAL	IIBAL	EFAL
		RFOL	PERRL	CCLL	CCFL	SFOL	SFBML	CNAL	PEMCL
	FRSILS	—	—	—	—	—	—	MTSBL	WUPBL
		—	—	—	—	—	—	MTSAL	WUPAL
		SDSL	MBSIL	SUCSL	SWEL	TOBCL	TIBCL	TI1L	TI0L
		NMVCL	RFCLL	RFNEL	RXIL	TXIL	CYCSL	CASL	WSTL
	FREIES	—	—	—	—	—	TABBE	LTVBE	EDBE
		—	—	—	—	—	TABAE	LTVAE	EDAE
		—	—	—	—	MHFE	IOBAE	IIBAE	EFAE
		RFOE	PERRE	CCLE	CCFE	SFOE	SFBME	CNAE	PEMCE
	FREIER	—	—	—	—	—	TABBE	LTVBE	EDBE
		—	—	—	—	—	TABAE	LTVAE	EDAE
		—	—	—	—	MHFE	IOBAE	IIBAE	EFAE
		RFOE	PERRE	CCLE	CCFE	SFOE	SFBME	CNAE	PEMCE
	FRSIES	—	—	—	—	—	—	MTSBE	WUPBE
		—	—	—	—	—	—	MTSAE	WUPAE
		SDSE	MBSIE	SUCSE	SWEE	TOBCE	TIBCE	TI1E	TI0E
		NMVCE	RFCLE	RFNEE	RXIE	TXIE	CYCSE	CASE	WSTE
	FRSIER	—	—	—	—	—	—	MTSBE	WUPBE
		—	—	—	—	—	—	MTSAE	WUPAE
		SDSE	MBSIE	SUCSE	SWEE	TOBCE	TIBCE	TI1E	TI0E
		NMVCE	RFCLE	RFNEE	RXIE	TXIE	CYCSE	CASE	WSTE

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
FlexRay	FRILE	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	EINT1	EINT0
	FRT0C	—	—	T0MO[13:8]					
		T0MO[7:0]							
		—	T0CC[6:0]						
		—	—	—	—	—	—	T0MS	T0RC
	FRT1C	—	—	T1MC[13:8]					
		T1MC[7:0]							
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	T1MS	T1RC
	FRSTPW1	—	—	SMTV[13:8]					
		SMTV[7:0]							
		—	—	SCCV[5:0]					
		—	EINT1	EINT0	—	SSWT	—	SWMS	ESWT
	FRSTPW2	—	—	—	—	—	SSCVB[10:8]		
		SSCVB[7:0]							
		—	—	—	—	—	SSCVA[10:8]		
		SSCVA[7:0]							
	FRSUCC1	—	—	—	—	CCHB	CCHA	MTSB	MTSA
		HCSE	TSM	WUCS	PTA[4:0]				
		CSA[4:0]					—	TXSY	TXST
		PBSY	—	—	—	CMD[3:0]			
	FRSUCC2	—	—	—	—	LTN[3:0]			
		—	—	—	LT[20:16]				
		LT[15:8]							
		LT[7:0]							
	FRSUCC3	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		WCF[3:0]				WCP[3:0]			

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
FlexRay	FRNEMC	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	NML[3:0]			
	FRPRTC1	RWP[5:0]						—	RXW[8]
		RXW[7:0]							
		—	BRP0	SPP1	SPP0	—	CASM6	CASM[5:4]	
		CASM[3:0]				TSST[3:0]			
	FRPRTC2	—	—	TXL[5:0]					
		TXI[7:0]							
		—	—	RXL[5:0]					
		—	—	RXI[5:0]					
	FRMHDC	—	—	—	SLT[12:8]				
		SLT[7:0]							
		—	—	—	—	—	—	—	—
		—	SFDL[6:0]						
	FRGTUC1	—	—	—	—	—	—	—	—
		—	—	—	—	UT[19:16]			
		UT[15:8]							
		UT[7:0]							
	FRGTUC2	—	—	—	—	—	—	—	—
		—	—	—	—	SNM[3:0]			
		—	—	MPC[13:8]					
		MPC[7:0]							
	FRGTUC3	—	MIOB[6:0]						
		—	MIOA[6:0]						
		UIOB[7:0]							
		UIOA[7:0]							
	FRGTUC4	—	—	OCS[13:8]					
		OCS[7:0]							
		—	—	NIT[13:8]					
		NIT[7:0]							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
FlexRay	FRGTUC5	DEC[7:0]							
		—	—	—	CDD[4:0]				
		DCB[7:0]							
		DCA[7:0]							
	FRGTUC6	—	—	—	—	—	MOD[10:8]		
		MOD[7:0]							
		—	—	—	—	—	ASR[10:8]		
		ASR[7:0]							
	FRGTUC7	—	—	—	—	—	—	NSS[9:8]	
		NSS[7:0]							
		—	—	—	—	—	—	SSL[9:8]	
		SSL[7:0]							
	FRGTUC8	—	—	—	NMS[12:8]				
		NMS[7:0]							
		—	—	—	—	—	—	—	—
		—	—	MSL[5:0]					
	FRGTUC9	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	DSI1	DSI0
		—	—	—	MAPO[4:0]				
		—	—	APO[5:0]					
	FRGTUC10	—	—	—	—	—	MRC[10:8]		
		MRC[7:0]							
		—	—	MOC[13:8]					
		MOC[7:0]							
	FRGTUC11	—	—	—	—	—	ERC[2:0]		
		—	—	—	—	—	EOC[2:0]		
		—	—	—	—	—	—	ERCC1	ERCC0
		—	—	—	—	—	—	EOCC1	EOCC0
	FRCCSV	—	—	PSL[5:0]					
		RCA[4:0]					WSV[2:0]		
		—	CSI	CSAI	CSNI	—	—	SLM1	SLM0
		HRQ	FSI	POCS[5:0]					

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
FlexRay	FRCCEV	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	PTAC[4:0]				
		ERRM1	ERRM0	—	—	CCFC[3:0]			
	FRSCV	—	—	—	—	—	SCCB[10:8]		
		SCCB[7:0]							
		—	—	—	—	—	SCCA[10:8]		
		SCCA[7:0]							
	FRMTCCV	—	—	—	—	—	—	—	—
		—	—	CCV[5:0]					
		—	—	MTV[13:8]					
		MTV[7:0]							
	FRRCV	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	RCV[11:8]			
		RCV[7:0]							
	FROCV	—	—	—	—	—	—	—	—
		—	—	—	—	—	OCV[18:16]		
		OCV[15:8]							
		OCV[7:0]							
	FRSFS	—	—	—	—	—	—	—	—
		—	—	—	—	RCLR	MRCS	OCLR	MOCS
		VSBO[3:0]				VSBE[3:0]			
		VSAO[3:0]				VSAE[3:0]			
	FRSWNIT	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	SBNB	SENB	SBNA	SENA
		MTSB	MTSA	TCSB	SBSB	SESB	TCSA	SBSA	SESA
	FRACS	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	SBVB	CIB	CEDB	SEDB	VFRB
		—	—	—	SBVA	CIA	CEDA	SEDA	VFRA

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
FlexRay	FRESID1	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		RXEB	RXEA	—	—	—	—	EID[9:8]	
		EID[7:0]							
	FRESID2	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		RXEB	RXEA	—	—	—	—	EID[9:8]	
		EID[7:0]							
	FRESID3	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		RXEB	RXEA	—	—	—	—	EID[9:8]	
		EID[7:0]							
	FRESID4	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		RXEB	RXEA	—	—	—	—	EID[9:8]	
		EID[7:0]							
	FRESID5	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		RXEB	RXEA	—	—	—	—	EID[9:8]	
		EID[7:0]							
	FRESID6	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		RXEB	RXEA	—	—	—	—	EID[9:8]	
		EID[7:0]							
	FRESID7	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		RXEB	RXEA	—	—	—	—	EID[9:8]	
		EID[7:0]							
	FRESID8	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		RXEB	RXEA	—	—	—	—	EID[9:8]	
		EID[7:0]							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
FlexRay	FRESID9	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		RXEB	RXEA	—	—	—	—	EID[9:8]	
		EID[7:0]							
	FRESID10	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		RXEB	RXEA	—	—	—	—	EID[9:8]	
		EID[7:0]							
	FRESID11	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		RXEB	RXEA	—	—	—	—	EID[9:8]	
		EID[7:0]							
	FRESID12	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		RXEB	RXEA	—	—	—	—	EID[9:8]	
		EID[7:0]							
	FRESID13	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		RXEB	RXEA	—	—	—	—	EID[9:8]	
		EID[7:0]							
	FRESID14	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		RXEB	RXEA	—	—	—	—	EID[9:8]	
		EID[7:0]							
	FRESID15	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		RXEB	RXEA	—	—	—	—	EID[9:8]	
		EID[7:0]							
	FROSID1	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		RXOB	RXOA	—	—	—	—	OID[9:8]	
		OID[7:0]							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
FlexRay	FROSID2	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		RXOB	RXOA	—	—	—	—	OID[9:8]	
		OID[7:0]							
	FROSID3	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		RXOB	RXOA	—	—	—	—	OID[9:8]	
		OID[7:0]							
	FROSID4	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		RXOB	RXOA	—	—	—	—	OID[9:8]	
		OID[7:0]							
	FROSID5	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		RXOB	RXOA	—	—	—	—	OID[9:8]	
		OID[7:0]							
	FROSID6	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		RXOB	RXOA	—	—	—	—	OID[9:8]	
		OID[7:0]							
	FROSID7	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		RXOB	RXOA	—	—	—	—	OID[9:8]	
		OID[7:0]							
	FROSID8	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		RXOB	RXOA	—	—	—	—	OID[9:8]	
		OID[7:0]							
	FROSID9	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		RXOB	RXOA	—	—	—	—	OID[9:8]	
		OID[7:0]							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
FlexRay	FROSID10	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		RXOB	RXOA	—	—	—	—	OID[9:8]	
		OID[7:0]							
	FROSID11	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		RXOB	RXOA	—	—	—	—	OID[9:8]	
		OID[7:0]							
	FROSID12	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		RXOB	RXOA	—	—	—	—	OID[9:8]	
		OID[7:0]							
	FROSID13	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		RXOB	RXOA	—	—	—	—	OID[9:8]	
		OID[7:0]							
	FROSID14	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		RXOB	RXOA	—	—	—	—	OID[9:8]	
		OID[7:0]							
	FROSID15	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		RXOB	RXOA	—	—	—	—	OID[9:8]	
		OID[7:0]							
	FRNMV1	NM[31:24]							
		NM[23:16]							
		NM[15:8]							
		NM[7:0]							
	FRNMV2	NM[31:24]							
		NM[23:16]							
		NM[15:8]							
		NM[7:0]							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
FlexRay	FRNMV3	NM[31:24]							
		NM[23:16]							
		NM[15:8]							
		NM[7:0]							
	FRMRC	—	—	—	—	—	SPLM	SEC1	SEC0
		LCB[7:0]							
		FFB[7:0]							
		FDB[7:0]							
	FRFRF	—	—	—	—	—	—	—	RNF
		RSS	CYF[6:0]						
		—	—	—	FID[10:6]				
		FID[5:0]						CH1	CH0
	FRFRFM	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	MFID[10:6]				
		MFID[5:0]						—	—
	FRFCL	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		CL[7:0]							
	FRMHDS	—	MBU[6:0]						
		—	MBT[6:0]						
		—	FMB[6:0]						
		CRAM	MFMB	FMBD	PTBF2	PTBF1	PMR	POBF	PIBF
	FRLDTS	—	—	—	—	—	LDTB[10:8]		
		LDTB[7:0]							
		—	—	—	—	—	LDTA[10:8]		
		LDTA[7:0]							
	FRFSR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		RFFL[7:0]							
		—	—	—	—	—	RFO	RFCL	RFNE

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
FlexRay	FRMHDF	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	WAHP
		—	—	TBFB	TBFA	FNFB	FNFA	SNUB	SNUA
	FRTXRQ1	TXR[31:24]							
		TXR[23:16]							
		TXR[15:8]							
		TXR[7:0]							
	FRTXRQ2	TXR[63:56]							
		TXR[55:48]							
		TXR[47:40]							
		TXR[39:32]							
	FRTXRQ3	TXR[95:88]							
		TXR[87:80]							
		TXR[79:72]							
		TXR[71:64]							
	FRTXRQ4	TXR[127:120]							
		TXR[119:112]							
		TXR[111:104]							
		TXR[103:96]							
	FRNDAT1	ND[31:24]							
		ND[23:16]							
		ND[15:8]							
		ND[7:0]							
	FRNDAT2	ND[63:56]							
		ND[55:48]							
		ND[47:40]							
		ND[39:32]							
	FRNDAT3	ND[95:88]							
		ND[87:80]							
		ND[79:72]							
		ND[71:64]							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
FlexRay	FRNDAT4	ND[127:120]							
		ND[119:112]							
		ND[111:104]							
		ND[103:96]							
	FRMBSC1	MBC[31:24]							
		MBC[23:16]							
		MBC[15:8]							
		MBC[7:0]							
	FRMBSC2	MBC[63:56]							
		MBC[55:48]							
		MBC[47:40]							
		MBC[39:32]							
	FRMBSC3	MBC[95:88]							
		MBC[87:80]							
		MBC[79:72]							
		MBC[71:64]							
	FRMBSC4	MBC[127:120]							
		MBC[119:112]							
		MBC[111:104]							
		MBC[103:96]							
	FRWRDS1	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS2	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS3	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
FlexRay	FRWRDS4	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS5	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS6	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS7	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS8	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS9	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS10	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS11	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
FlexRay	FRWRDS12	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS13	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS14	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS15	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS16	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS17	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS18	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS19	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							

Module	**Register Name**	**31/23/15/7**	**30/22/14/6**	**29/21/13/5**	**28/20/12/4**	**27/19/11/3**	**26/18/10/2**	**25/17/9/1**	**24/16/8/0**
FlexRay	FRWRDS20	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS21	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS22	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS23	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS24	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS25	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS26	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS27	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
FlexRay	FRWRDS28	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS29	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS30	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS31	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS32	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS33	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS34	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS35	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
FlexRay	FRWRDS36	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS37	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS38	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS39	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS40	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS41	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS42	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS43	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
FlexRay	FRWRDS44	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS45	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS46	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS47	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS48	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS49	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS50	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS51	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
FlexRay	FRWRDS52	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS53	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS54	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS55	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS56	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS57	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS58	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS59	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
FlexRay	FRWRDS60	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS61	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS62	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS63	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRDS64	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRWRHS1	—	—	MBI	TXM	PPIT	CFG	CHB	CHA
		—	CYC[6:0]						
		—	—	—	—	—	FID[10:8]		
		FID[7:0]							
	FRWRHS2	—	—	—	—	—	—	—	—
		—	PLC[6:0]						
		—	—	—	—	—	CRC[10:8]		
		CRC[7:0]							
	FRWRHS3	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	DP[10:8]		
		DP[7:0]							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
FlexRay	FRIBCM	—	—	—	—	—	—	—	—
		—	—	—	—	—	STXRS	LDSS	LHSS
		—	—	—	—	—	—	—	—
		—	—	—	—	—	STXRH	LDSH	LHSH
	FRIBCR	IBSYS	—	—	—	—	—	—	—
		—	IBRS[6:0]						
		IBSYH	—	—	—	—	—	—	—
		—	IBRH[6:0]						
	FRRDDS1	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS2	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS3	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS4	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS5	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS6	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
FlexRay	FRRDDS7	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS8	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS9	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS10	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS11	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS12	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS13	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS14	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
FlexRay	FRRDDS15	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS16	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS17	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS18	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS19	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS20	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS21	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS22	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
FlexRay	FRRDDS23	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS24	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS25	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS26	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS27	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS28	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS29	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS30	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
FlexRay	FRRDDS31	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS32	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS33	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS34	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS35	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS36	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS37	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS38	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
FlexRay	FRRDDS39	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS40	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS41	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS42	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS43	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS44	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS45	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS46	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
FlexRay	FRRDDS47	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS48	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS49	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS50	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS51	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS52	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS53	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS54	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
FlexRay	FRRDDS55	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS56	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS57	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS58	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS59	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS60	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS61	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS62	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
FlexRay	FRRDDS63	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDDS64	MD[31:24]							
		MD[23:16]							
		MD[15:8]							
		MD[7:0]							
	FRRDHS1	—	—	MBI	TXM	PPIT	CFG	CHB	CHA
		—	CYC[6:0]						
		—	—	—	—	—	FID[10:8]		
		FID[7:0]							
	FRRDHS2	—	PLR[6:0]						
		—	PLC[6:0]						
		—	—	—	—	—	CRC[10:8]		
		CRC[7:0]							
	FRRDHS3	—	—	RES	PPI	NFI	SYN	SFI	RCI
		—	—	RCC[5:0]					
		—	—	—	—	—	DP[10:8]		
		DP[7:0]							
	FRMBS	—	—	RESS	PPIS	NFIS	SYNS	SFIS	RCIS
		—	—	CCS[5:0]					
		FTB	FTA	—	MLST	ESB	ESA	TCIB	TCIA
		SVOB	SVOA	CEOB	CEOA	SEOB	SEOA	VFRB	VFRA
	FROBCM	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	RDSH	RHSH
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	RDSS	RHSS
	FROBCR	—	—	—	—	—	—	—	—
		—	OBRH[6:0]						
		OBSYS	—	—	—	—	—	REQ	VIEW
		—	OBRS[6:0]						

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ADC (ADC_A)	ADRD0								
	ADR0								
	ADR1								
	ADR2								
	ADR3								
	ADR4								
	ADR5								
	ADR6								
	ADR7								
	ADR8								
	ADR9								
	ADR10								
	ADR11								
	ADR12								
	ADR13								
	ADR14								

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ADC (ADC_A)	ADR15								
	ADR16								
	ADR17								
	ADR18								
	ADR19								
	ADR20								
	ADR21								
	ADR22								
	ADR23								
	ADR24								
	ADR25								
	ADR26								
	ADR27								
ADC (ADC_B)	ADRD1								
	ADR40								
	ADR41								

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ADC (ADC_B)	ADR42								
	ADR43								
	ADR44								
	ADR45								
	ADR46								
	ADR47								
	ADR48								
ADC (ADC_A)	ADCSR0	ADST	ADCS	—	ADIE	—	—	TRGE	EXTRG
ADC (ADC_B)	ADCSR1	ADST	ADCS	—	ADIE	—	—	TRGE	EXTRG
ADC (ADC_A)	ADREF0	ADSCACT	ADITACT	—	—	—	—	—	ADF
ADC (ADC_B)	ADREF1	ADSCACT	ADITACT	—	—	—	—	—	ADF
ADC (ADC_A)	ADTRE0	ADTRGE15	ADTRGE14	ADTRGE13	ADTRGE12	ADTRGE11	ADTRGE10	ADTRGE9	ADTRGE8
		ADTRGE7	ADTRGE6	ADTRGE5	ADTRGE4	ADTRGE3	ADTRGE2	ADTRGE1	ADTRGE0
ADC (ADC_B)	ADTRE1	ADTRGE47	ADTRGE46	ADTRGE45	ADTRGE44	ADTRGE43	ADTRGE42	ADTRGE41	ADTRGE40
ADC (ADC_A)	ADTRF0	ADTF15	ADTF14	ADTF13	ADTF12	ADTF11	ADTF10	ADTF9	ADTF8
		ADTF7	ADTF6	ADTF5	ADTF4	ADTF3	ADTF2	ADTF1	ADTF0
ADC (ADC_B)	ADTRF1	ADTF47	ADTF46	ADTF45	ADTF44	ADTF43	ADTF42	ADTF41	ADTF40
ADC (ADC_A)	ADTRS0	ADTRS15	ADTRS14	ADTRS13	ADTRS12	ADTRS11	ADTRS10	ADTRS9	ADTRS8
		ADTRS7	ADTRS6	ADTRS5	ADTRS4	ADTRS3	ADTRS2	ADTRS1	ADTRS0
ADC (ADC_B)	ADTRS1	ADTRS47	ADTRS46	ADTRS45	ADTRS44	ADTRS43	ADTRS42	ADTRS41	ADTRS40

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ADC (ADC_A)	ADSTRG0	ADSTRG15	ADSTRG14	ADSTRG13	ADSTRG12	ADSTRG11	ADSTRG10	ADSTRG9	ADSTRG8
		ADSTRG7	ADSTRG6	ADSTRG5	ADSTRG4	ADSTRG3	ADSTRG2	ADSTRG1	ADSTRG0
ADC (ADC_B)	ADSTRG1	ADSTRG47	ADSTRG46	ADSTRG45	ADSTRG44	ADSTRG43	ADSTRG42	ADSTRG41	ADSTRG40
ADC (ADC_A)	ADTRD0	ADIDE15	ADIDE14	ADIDE13	ADIDE12	ADIDE11	ADIDE10	ADIDE9	ADIDE8
		ADIDE7	ADIDE6	ADIDE5	ADIDE4	ADIDE3	ADIDE2	ADIDE1	ADIDE0
ADC (ADC_B)	ADTRD1	ADIDE47	ADIDE46	ADIDE45	ADIDE44	ADIDE43	ADIDE42	ADIDE41	ADIDE40
ADC (ADC_A)	ADADS0	ADS7	ADS6	ADS5	ADS4	ADS3	ADS2	ADS1	ADS0
	ADADS2	ADS15	ADS14	ADS13	ADS12	ADS11	ADS10	ADS9	ADS8
	ADADS3	ADS23	ADS22	ADS21	ADS20	ADS19	ADS18	ADS17	ADS16
	ADADS4	—	—	—	—	ADS27	ADS26	ADS25	ADS24
ADC (ADC_B)	ADADS1	ADS47	ADS46	ADS45	ADS44	ADS43	ADS42	ADS41	ADS40
	ADADS5	—	—	—	—	—	—	—	ADS48
ADC (ADC_A)	ADADC0	—	—	—	—	—	—	ADC[1:0]	
ADC (ADC_B)	ADADC1	—	—	—	—	—	—	ADC[1:0]	
ADC (ADC_A)	ADANS0	ANS15	ANS14	ANS13	ANS12	ANS11	ANS10	ANS9	ANS8
		ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0
	ADANS1	—	—	—	ANS28	ANS27	ANS26	ANS25	ANS24
		ANS23	ANS22	ANS21	ANS20	ANS19	ANS18	ANS17	ANS16
ADC (ADC_B)	ADANS3	—	—	—	—	—	—	—	ANS48
		ANS47	ANS46	ANS45	ANS44	ANS43	ANS42	ANS41	ANS40
ADC (ADC_A)	ADCER0	ADRFMT	DIAGSEL	DIAGVALE	DIAGVAL2	DIAGM	DIAGLD	DIAGVAL[1:0]	
		CKS	—	—	—	—	—	—	ITTRGS
ADC (ADC_B)	ADCER1	ADRFMT	DIAGSEL	DIAGVALE	DIAGVAL2	DIAGM	DIAGLD	DIAGVAL[1:0]	
		CKS	—	—	—	—	—	—	—
JTAG	SDIR								
	SDID								

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
JTAG	SDBPR								
	SDBSR								
AUD-II	AUCSR	CLK[1:0]		BW[1:0]		OC[1:0]		BR[1:0]	
		WA[1:0]		WB[1:0]		—	TM	—	EN
	AUWASR								
	AUWAER								
	AUWBSR								
	AUWBER								
	AUECSR	—	—	—	—	—	—	WA0B[2:1]	
		WA0B[0]	WB0B[2:0]			TREX	TRSB	TRGN	—
PFC	PAIOR	PA15IOR	PA14IOR	PA13IOR	PA12IOR	PA11IOR	PA10IOR	PA9IOR	PA8IOR
		PA7IOR	PA6IOR	PA5IOR	PA4IOR	PA3IOR	PA2IOR	PA1IOR	PA0IOR
	PACR4	—	—	—	PA15MD	—	—	—	PA14MD
		—	—	—	PA13MD	—	—	—	PA12MD
	PACR3	—	—	—	PA11MD	—	—	—	PA10MD
		—	—	—	PA9MD	—	—	—	PA8MD
	PACR2	—	—	—	PA7MD	—	—	—	PA6MD
		—	—	—	PA5MD	—	—	—	PA4MD

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
PFC	PACR1	—	—	—	PA3MD	—	—	—	PA2MD
		—	—	—	PA1MD	—	—	—	PA0MD
	PBIOR	—	PB14IOR	PB13IOR	PB12IOR	PB11IOR	PB10IOR	PB9IOR	PB8IOR
		PB7IOR	PB6IOR	PB5IOR	PB4IOR	PB3IOR	PB2IOR	PB1IOR	PB0IOR
	PBCR4	—	—	—	—	—	PB14MD[2:0]		
		—	—	PB13MD[1:0]		—	—	PB12MD[1:0]	
	PBCR3	—	—	PB11MD[1:0]		—	—	—	PB10MD
		—	—	—	PB9MD	—	—	PB8MD[1:0]	
	PBCR2	—	—	—	PB7MD	—	—	—	PB6MD
		—	PB5MD[2:0]			—	PB4MD[2:0]		
	PBCR1	—	—	PB3MD[1:0]		—	—	PB2MD[1:0]	
		—	—	PB1MD[1:0]		—	—	PB0MD[1:0]	
	PCIOR	PC15IOR	PC14IOR	PC13IOR	PC12IOR	PC11IOR	PC10IOR	PC9IOR	PC8IOR
		PC7IOR	PC6IOR	PC5IOR	PC4IOR	PC3IOR	PC2IOR	PC1IOR	PC0IOR
	PCCR4	—	—	—	PC15MD	—	—	—	PC14MD
		—	—	—	PC13MD	—	—	—	PC12MD
	PCCR3	—	—	—	PC11MD	—	—	—	PC10MD
		—	—	—	PC9MD	—	—	—	PC8MD
	PCCR2	—	—	—	PC7MD	—	—	—	PC6MD
		—	—	—	PC5MD	—	—	—	PC4MD
	PCCR1	—	—	—	PC3MD	—	—	—	PC2MD
		—	—	—	PC1MD	—	—	—	PC0MD
	PDIOR	—	—	PD13IOR	PD12IOR	PD11IOR	PD10IOR	PD9IOR	PD8IOR
		PD7IOR	PD6IOR	PD5IOR	PD4IOR	PD3IOR	PD2IOR	PD1IOR	PD0IOR
	PDCR2	—	—	—	—	PD13MD[1:0]		PD12MD[1:0]	
		PD11MD[1:0]		PD10MD[1:0]		PD9MD[1:0]		PD8MD[1:0]	
	PDCR1	PD7MD[1:0]		PD6MD[1:0]		PD5MD[1:0]		PD4MD[1:0]	
		PD3MD[1:0]		PD2MD[1:0]		PD1MD[1:0]		PD0MD[1:0]	
	PEIOR	—	—	PE13IOR	PE12IOR	PE11IOR	PE10IOR	PE9IOR	PE8IOR
		PE7IOR	PE6IOR	PE5IOR	PE4IOR	PE3IOR	PE2IOR	PE1IOR	PE0IOR
	PECR2	—	—	—	—	—	PE13MD	—	PE12MD
		—	PE11MD	—	PE10MD	—	PE9MD	—	PE8MD

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
PFC	PECR1	PE7MD[1:0]		PE6MD[1:0]		—	PE5MD	—	PE4MD
		—	PE3MD	PE2MD[1:0]		PE1MD[1:0]		—	PE0MD
	PFIOR	PF15IOR	PF14IOR	PF13IOR	PF12IOR	PF11IOR	PF10IOR	PF9IOR	PF8IOR
		PF7IOR	PF6IOR	PF5IOR	PF4IOR	PF3IOR	PF2IOR	PF1IOR	PF0IOR
	PFCR2	PF15MD[1:0]		PF14MD[1:0]		PF13MD[1:0]		PF12MD[1:0]	
		PF11MD[1:0]		PF10MD[1:0]		PF9MD[1:0]		PF8MD[1:0]	
	PFCR1	PF7MD[1:0]		PF6MD[1:0]		PF5MD[1:0]		PF4MD[1:0]	
		PF3MD[1:0]		PF2MD[1:0]		PF1MD[1:0]		PF0MD[1:0]	
	PGIOR	PG15IOR	PG14IOR	PG13IOR	PG12IOR	PG11IOR	PG10IOR	PG9IOR	PG8IOR
		PG7IOR	PG6IOR	PG5IOR	PG4IOR	PG3IOR	PG2IOR	PG1IOR	PG0IOR
	PGCR2	PG15MD[1:0]		PG14MD[1:0]		PG13MD[1:0]		PG12MD[1:0]	
		PG11MD[1:0]		PG10MD[1:0]		PG9MD[1:0]		PG8MD[1:0]	
	PGCR1	PG7MD[1:0]		PG6MD[1:0]		PG5MD[1:0]		PG4MD[1:0]	
		PG3MD[1:0]		PG2MD[1:0]		PG1MD[1:0]		PG0MD[1:0]	
	PHIOR	—	—	—	—	—	—	—	—
		—	—	PH5IOR	PH4IOR	PH3IOR	PH2IOR	PH1IOR	PH0IOR
	PHCR	—	—	—	—	PH5MD[1:0]		PH4MD[1:0]	
		PH3MD[1:0]		PH2MD[1:0]		PH1MD[1:0]		PH0MD[1:0]	
	PJIOR	—	—	—	—	—	—	PJ9IOR	PJ8IOR
		PJ7IOR	PJ6IOR	PJ5IOR	PJ4IOR	PJ3IOR	PJ2IOR	PJ1IOR	PJ0IOR
	PJCR2	—	—	—	—	—	—	—	—
		—	—	—	—	—	PJ9MD	—	PJ8MD
	PJCR1	PJ7MD[1:0]		—	PJ6MD	—	PJ5MD	PJ4MD[1:0]	
		PJ3MD[1:0]		PJ2MD[1:0]		PJ1MD[1:0]		PJ0MD[1:0]	
	PKIOR	—	—	—	—	PK11IOR	PK10IOR	PK9IOR	PK8IOR
		PK7IOR	PK6IOR	PK5IOR	PK4IOR	PK3IOR	PK2IOR	PK1IOR	PK0IOR
	PKCR2	—	—	—	—	—	—	—	—
		PK11MD[1:0]		PK10MD[1:0]		PK9MD[1:0]		—	PK8MD
	PKCR1	—	PK7MD	—	PK6MD	PK5MD[1:0]		PK4MD[1:0]	
		PK3MD[1:0]		PK2MD[1:0]		PK1MD[1:0]		PK0MD[1:0]	
	PLIOR	—	—	—	—	—	—	—	PL8IOR
		PL7IOR	PL6IOR	PL5IOR	PL4IOR	PL3IOR	PL2IOR	PL1IOR	PL0IOR

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
PFC	PLCR2	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	PL8MD
	PLCR1	PL7MD[1:0]		PL6MD[1:0]		PL5MD[1:0]		PL4MD[1:0]	
		PL3MD[1:0]		PL2MD[1:0]		PL1MD[1:0]		PL0MD[1:0]	
	PACR4A	—	—	—	PA15AMD	—	—	—	PA14AMD
		—	—	—	PA13AMD	—	—	—	PA12AMD
	PACR3A	—	—	PA11AMD[1:0]		—	—	PA10AMD[1:0]	
		—	—	PA9AMD[1:0]		—	—	—	PA8AMD
	PACR2A	—	—	PA7AMD[1:0]		—	—	PA6AMD[1:0]	
		—	—	PA5AMD[1:0]		—	—	—	PA4AMD
	PBCR4A	—	—	—	—	—	—	—	—
		—	—	—	PB13AMD	—	—	PB12AMD[1:0]	
	PBCR3A	—	—	PB11AMD[1:0]		—	—	PB10AMD[1:0]	
		—	—	PB9AMD[1:0]		—	—	PB8AMD[1:0]	
	PBCR2A	—	—	PB7AMD[1:0]		—	—	PB6AMD[1:0]	
		—	—	PB5AMD[1:0]		—	—	PB4AMD[1:0]	
	PBCR1A	—	—	PB3AMD[1:0]		—	—	PB2AMD[1:0]	
		—	—	PB1AMD[1:0]		—	—	PB0AMD[1:0]	
	PCCR4A	—	—	PC15AMD[1:0]		—	—	PC14AMD[1:0]	
		—	—	PC13AMD[1:0]		—	—	PC12AMD[1:0]	
	PCCR3A	—	—	PC11AMD[1:0]		—	—	PC10AMD[1:0]	
		—	—	PC9AMD[1:0]		—	—	PC8AMD[1:0]	
	PCCR2A	—	—	PC7AMD[1:0]		—	—	PC6AMD[1:0]	
		—	—	—	—	—	—	—	—
	PDCR2A	—	—	—	—	—	PD13AMD	—	—
		—	—	—	—	—	—	—	—
	PECR1A	—	—	—	—	PE5AMD[1:0]		PE4AMD[1:0]	
		PE3AMD[1:0]		—	—	—	—	—	—
	PFCR2A	PF15AMD[1:0]		PF14AMD[1:0]		—	—	—	—
		—	—	—	—	—	—	—	—
	PJCR3A	—	—	—	—	—	—	—	—
		—	—	PJ9AMD[1:0]		—	—	PJ8AMD[1:0]	

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
PFC	PJCR2A	—	—	PJ7AMD[1:0]		—	—	—	—
		—	—	—	—	—	—	—	—
	PJCR1A	—	—	PJ3AMD[1:0]		—	—	PJ2AMD[1:0]	
		—	—	PJ1AMD[1:0]		—	—	PJ0AMD[1:0]	
I/O port	PADR	PA15DR	PA14DR	PA13DR	PA12DR	PA11DR	PA10DR	PA9DR	PA8DR
		PA7DR	PA6DR	PA5DR	PA4DR	PA3DR	PA2DR	PA1DR	PA0DR
	PAPR	PA15PR	PA14PR	PA13PR	PA12PR	PA11PR	PA10PR	PA9PR	PA8PR
		PA7PR	PA6PR	PA5PR	PA4PR	PA3PR	PA2PR	PA1PR	PA0PR
	PAIR	—	—	—	—	PA11IR	PA10IR	PA9IR	PA8IR
		PA7IR	PA6IR	PA5IR	PA4IR	—	—	—	—
	PBDR	—	PB14DR	PB13DR	PB12DR	PB11DR	PB10DR	PB9DR	PB8DR
		PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR
	PBPR	—	PB14PR	PB13PR	PB12PR	PB11PR	PB10PR	PB9PR	PB8PR
		PB7PR	PB6PR	PB5PR	PB4PR	PB3PR	PB2PR	PB1PR	PB0PR
	PBIR	—	—	PB13IR	PB12IR	PB11IR	PB10IR	PB9IR	PB8IR
		PB7IR	PB6IR	—	PB4IR	PB3IR	PB2IR	PB1IR	PB0IR
	PBDSR	—	—	PB13DSR	PB12DSR	PB11DSR	PB10DSR	PB9DSR	—
		PB7DSR	PB6DSR	—	PB4DSR	PB3DSR	PB2DSR	PB1DSR	PB0DSR
	PBPSR	—	—	—	—	—	—	—	PB8PSR
		—	—	PB5PSR	—	PB3PSR	—	PB1PSR	—
	PCDR	PC15DR	PC14DR	PC13DR	PC12DR	PC11DR	PC10DR	PC9DR	PC8DR
		PC7DR	PC6DR	PC5DR	PC4DR	PC3DR	PC2DR	PC1DR	PC0DR
	PCPR	PC15PR	PC14PR	PC13PR	PC12PR	PC11PR	PC10PR	PC9PR	PC8PR
		PC7PR	PC6PR	PC5PR	PC4PR	PC3PR	PC2PR	PC1PR	PC0PR
	PCIR	PC15IR	PC14IR	PC13IR	PC12IR	PC11IR	PC10IR	PC9IR	PC8IR
		—	—	—	—	—	—	—	—
	PCDSR	PC15DSR	PC14DSR	PC13DSR	PC12DSR	PC11DSR	PC10DSR	—	PC8DSR
		—	—	—	—	—	—	—	—
	PCPSR	—	—	—	—	—	—	PC9PSR	—
		—	—	—	—	—	—	—	—
	PDDR	—	—	PD13DR	PD12DR	PD11DR	PD10DR	PD9DR	PD8DR
		PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DR

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
I/O port	PDPR	—	—	PD13PR	PD12PR	PD11PR	PD10PR	PD9PR	PD8PR
		PD7PR	PD6PR	PD5PR	PD4PR	PD3PR	PD2PR	PD1PR	PD0PR
	PDIR	—	—	PD13IR	PD12IR	PD11IR	PD10IR	PD9IR	PD8IR
		PD7IR	PD6IR	PD5IR	PD4IR	PD3IR	PD2IR	PD1IR	PD0IR
	PEDR	—	—	PE13DR	PE12DR	PE11DR	PE10DR	PE9DR	PE8DR
		PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR	PE0DR
	PEPR	—	—	PE13PR	PE12PR	PE11PR	PE10PR	PE9PR	PE8PR
		PE7PR	PE6PR	PE5PR	PE4PR	PE3PR	PE2PR	PE1PR	PE0PR
	PEIR	—	—	PE13IR	PE12IR	PE11IR	PE10IR	PE9IR	PE8IR
		PE7IR	PE6IR	PE5IR	PE4IR	PE3IR	PE2IR	PE1IR	—
	PEDSR	—	—	—	—	—	PE10DSR	PE9DSR	PE8DSR
		PE7DSR	PE6DSR	—	—	—	—	—	—
	PFDR	PF15DR	PF14DR	PF13DR	PF12DR	PF11DR	PF10DR	PF9DR	PF8DR
		PF7DR	PF6DR	PF5DR	PF4DR	PF3DR	PF2DR	PF1DR	PF0DR
	PFPR	PF15PR	PF14PR	PF13PR	PF12PR	PF11PR	PF10PR	PF9PR	PF8PR
		PF7PR	PF6PR	PF5PR	PF4PR	PF3PR	PF2PR	PF1PR	PF0PR
	PFIR	PF15IR	PF14IR	PF13IR	PF12IR	PF11IR	PF10IR	PF9IR	PF8IR
		PF7IR	PF6IR	PF5IR	PF4IR	PF3IR	PF2IR	PF1IR	PF0IR
	PFDSR	—	PF14DSR	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
	PFPSR	PF15PSR	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
	PGDR	PG15DR	PG14DR	PG13DR	PG12DR	PG11DR	PG10DR	PG9DR	PG8DR
		PG7DR	PG6DR	PG5DR	PG4DR	PG3DR	PG2DR	PG1DR	PG0DR
	PGPR	PG15PR	PG14PR	PG13PR	PG12PR	PG11PR	PG10PR	PG9PR	PG8PR
		PG7PR	PG6PR	PG5PR	PG4PR	PG3PR	PG2PR	PG1PR	PG0PR
	PGIR	PG15IR	PG14IR	PG13IR	PG12IR	PG11IR	PG10IR	PG9IR	PG8IR
		PG7IR	PG6IR	PG5IR	PG4IR	PG3IR	PG2IR	PG1IR	PG0IR
	PGDSR	PG15DSR	PG14DSR	PG13DSR	PG12DSR	PG11DSR	PG10DSR	PG9DSR	PG8DSR
		PG7DSR	PG6DSR	PG5DSR	PG4DSR	PG3DSR	PG2DSR	PG1DSR	PG0DSR
	PGER	—	—	—	—	—	—	PGHES[1:0]	
		—	—	—	—	—	—	PGLES[1:0]	

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
I/O port	PHDR	—	—	—	—	—	—	—	—
		—	—	PH5DR	PH4DR	PH3DR	PH2DR	PH1DR	PH0DR
	PHPR	—	—	—	—	—	—	—	—
		—	—	PH5PR	PH4PR	PH3PR	PH2PR	PH1PR	PH0PR
	PJDR	—	—	—	—	—	—	PJ9DR	PJ8DR
		PJ7DR	PJ6DR	PJ5DR	PJ4DR	PJ3DR	PJ2DR	PJ1DR	PJ0DR
	PJPR	—	—	—	—	—	—	PJ9PR	PJ8PR
		PJ7PR	PJ6PR	PJ5PR	PJ4PR	PJ3PR	PJ2PR	PJ1PR	PJ0PR
	PJIR	—	—	—	—	—	—	—	PJ8IR
		PJ7IR	—	PJ5IR	PJ4IR	—	PJ2IR	—	PJ0IR
	PJDSR	—	—	—	—	—	—	—	PJ8DSR
		PJ7DSR	—	PJ5DSR	PJ4DSR	—	PJ2DSR	—	PJ0DSR
	PJPSR	—	—	—	—	—	—	PJ9PSR	—
		—	PJ6PSR	—	—	PJ3PSR	—	PJ1PSR	—
	PKDR	—	—	—	—	PK11DR	PK10DR	PK9DR	PK8DR
		PK7DR	PK6DR	PK5DR	PK4DR	PK3DR	PK2DR	PK1DR	PK0DR
	PKPR	—	—	—	—	PK11PR	PK10PR	PK9PR	PK8PR
		PK7PR	PK6PR	PK5PR	PK4PR	PK3PR	PK2PR	PK1PR	PK0PR
	PKIR	—	—	—	—	PK11IR	PK10IR	PK9IR	—
		PK7IR	PK6IR	PK5IR	PK4IR	PK3IR	PK2IR	PK1IR	PK0IR
	PKDSR	—	—	—	—	PK11DSR	PK10DSR	PK9DSR	—
		PK7DSR	PK6DSR	PK5DSR	PK4DSR	PK3DSR	PK2DSR	PK1DSR	PK0DSR
	PKPSR	—	—	—	—	PK11PSR	—	—	PK8PSR
		—	—	PK5PSR	—	—	PK2PSR	—	—
	PLDR	—	—	—	—	—	—	—	PL8DR
		PL7DR	PL6DR	PL5DR	PL4DR	PL3DR	PL2DR	PL1DR	PL0DR
	PLPR	—	—	—	—	—	—	—	PL8PR
		PL7PR	PL6PR	PL5PR	PL4PR	PL3PR	PL2PR	PL1PR	PL0PR
	PLIR	—	—	—	—	—	—	—	PL8IR
		PL7IR	PL6IR	PL5IR	PL4IR	PL3IR	PL2IR	PL1IR	—
	CKCR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	CKOE

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
MISG	MISRCDR	MISRCD[31:24]							
		MISRCD[23:16]							
		MISRCD[15:8]							
		MISRCD[7:0]							
	MISR	MISR[31:24]							
		MISR[23:16]							
		MISR[15:8]							
		MISR[7:0]							
	MISR2	MISR2[31:24]							
		MISR2[23:16]							
		MISR2[15:8]							
		MISR2[7:0]							
	MISRCR	—	—	—	—	—	—	MISR2EN	MISREN
ROM/IFS/ EEPROM	FPMON	FWE	—	—	—	—	—	—	—
	FMODR	—	—	—	FRDMD	—	—	—	—
	FASTAT	ROMAE	—	—	CMDLK	EEPAE	EEPIFE	EEPRPE	EEPWPE
	FAEINT	ROMAEIE	—	—	CMDLKIE	EEPAEIE	EEPIFEIE	EEPRPEIE	EEPWPEIE
	ROMMAT	KEY							
		—	—	—	—	—	—	—	ROMSEL
	EEPRE0	KEY							
		DBRE07	DBRE06	DBRE05	DBRE04	DBRE03	DBRE02	DBRE01	DBRE00
	EEPRE1	KEY							
		DBRE15	DBRE14	DBRE13	DBRE12	DBRE11	DBRE10	DBRE09	DBRE08
	EEPWE0	KEY							
		DBWE07	DBWE06	DBWE05	DBWE04	DBWE03	DBWE02	DBWE01	DBWE00
	EEPWE1	KEY							
		DBWE15	DBWE14	DBWE13	DBWE12	DBWE11	DBWE10	DBWE09	DBWE08
	FCURAME	KEY							
		—	—	—	—	—	—	—	FCRME

Module	Register Name	31/23/15/7	30/22/14/6	29/21/13/5	28/20/12/4	27/19/11/3	26/18/10/2	25/17/9/1	24/16/8/0
ROM/IFS/EEPROM	IFSCMR	KEY							
		—	—	—	—	—	—	—	IFSCM
	FSTATR0	FRDY	ILGLERR	ERSERR	PRGERR	SUSRDY	—	ERSSPD	PRGSPD
	FSTATR1	FCUERR	—	—	FLOCKST	—	—	FRDTCT	FRCRCT
	FENTRYR	FEKEY							
		FENTRYD	FENTRY6	FENTRY5	FENTRY4	FENTRY3	FENTRY2	FENTRY1	FENTRY0
	FPROTR	FPKEY							
		—	—	—	—	—	—	—	FPROTCN
	FRESETR	FRKEY							
		—	—	—	—	—	—	—	FRESET
	FCMDR	CMDR							
		PCMDR							
	FRAMECCR	—	—	—	—	—	—	FRDCLE	FRCCLE
	IFSSR	—	—	—	—	—	IFSST[2:0]		
	FCPSR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	ESUSPMD
	EEPBCCNT	—	—	—	BCADR				
		BCADR					—	—	BCSIZE
	FPESTAT	—	—	—	—	—	—	—	—
		PEERRST							
	EEPBCSTAT	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	BCST
	EEPMAT	KEY							
		—	—	—	—	—	—	—	EEPSEL
ROMC	RCCR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	RCF	RCFI	RCFD	RCE
	RCCR2	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		PFECB	PFENB	PFECF	PFE	—	PCE2	—	PCE0

Module	**Register Name**	**31/23/15/7**	**30/22/14/6**	**29/21/13/5**	**28/20/12/4**	**27/19/11/3**	**26/18/10/2**	**25/17/9/1**	**24/16/8/0**
RAM	RAMEN0	RN0KEY[7:0]							
		RAME7	RAME6	RAME5	RAME4	RAME3	RAME2	RAME1	RAME0
	RAMWEN0	RWN0KEY[7:0]							
		RAMWE7	RAMWE6	RAMWE5	RAMWE4	RAMWE3	RAMWE2	RAMWE1	RAMWE0
	RAMECC	REKEY[7:0]							
		—	—	—	—	—	—	—	RECCA
	RAMERR	—	—	RPARI1	RPARI0	RDTCT1	RCRCT1	RDTCT0	RCRCT0
	RAMINT	—	—	RPEIE1	RPEIE0	REDIE1	RECIE1	REDIE0	RECIE0
	RAMACYC	RAKEY[7:0]							
		—	—	WRCYC[1:0]		—	—	—	RDCYC
	RAMEN1	RN1KEY[7:0]							
		RAME15	RAME14	RAME13	RAME12	RAME11	RAME10	RAME9	RAME8
	RAMWEN1	RWN1KEY[7:0]							
		RAMWE15	RAMWE14	RAMWE13	RAMWE12	RAMWE11	RAMWE10	RAMWE9	RAMWE8
Power-down mode	STBCR	STBCRKEY							
		—	—	MSTP5	MSTP4	MSTP3	MSTP2	MSTP1	MSTP0

33.3 Register States in Each Operating Mode

		Reset State	Power-Down State	
Module	**Register Name**	**Power-On**	**Hardware Standby**	**Sleep**
INTC	ICR0	Initialized	Initialized	Retained
	ICR1	Initialized	Initialized	Retained
	IRQRR	Initialized	Initialized	Retained
	IBCR	Initialized	Initialized	Retained
	IBNR	Initialized	Initialized	Retained
	SINTR1	Initialized	Initialized	Retained
	SINTR2	Initialized	Initialized	Retained
	SINTR3	Initialized	Initialized	Retained
	SINTR4	Initialized	Initialized	Retained
	SINTR5	Initialized	Initialized	Retained
	SINTR6	Initialized	Initialized	Retained
	SINTR7	Initialized	Initialized	Retained
	SINTR8	Initialized	Initialized	Retained
	IPR01	Initialized	Initialized	Retained
	IPR02	Initialized	Initialized	Retained
	SINTR9	Initialized	Initialized	Retained
	SINTR10	Initialized	Initialized	Retained
	SINTR11	Initialized	Initialized	Retained
	SINTR12	Initialized	Initialized	Retained
	SINTR13	Initialized	Initialized	Retained
	SINTR14	Initialized	Initialized	Retained
	SINTR15	Initialized	Initialized	Retained
	IPR03	Initialized	Initialized	Retained
	IPR04	Initialized	Initialized	Retained
	IPR05	Initialized	Initialized	Retained
	IPR06	Initialized	Initialized	Retained
	IPR07	Initialized	Initialized	Retained
	IPR08	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	Register Name	Power-On	Hardware Standby	Sleep
INTC	IPR09	Initialized	Initialized	Retained
	IPR10	Initialized	Initialized	Retained
	IPR11	Initialized	Initialized	Retained
	IPR12	Initialized	Initialized	Retained
	IPR13	Initialized	Initialized	Retained
	IPR14	Initialized	Initialized	Retained
	IPR15	Initialized	Initialized	Retained
	IPR16	Initialized	Initialized	Retained
	IPR17	Initialized	Initialized	Retained
	IPR18	Initialized	Initialized	Retained
	IPR19	Initialized	Initialized	Retained
	IPR20	Initialized	Initialized	Retained
	IPR21	Initialized	Initialized	Retained
	IPR22	Initialized	Initialized	Retained
	IPR23	Initialized	Initialized	Retained
	IPR24	Initialized	Initialized	Retained
	IPR25	Initialized	Initialized	Retained
	IPR26	Initialized	Initialized	Retained
	IPR27	Initialized	Initialized	Retained
	IPR28	Initialized	Initialized	Retained
	IPR29	Initialized	Initialized	Retained
	IPR30	Initialized	Initialized	Retained
UBC	BAR0	Initialized	Initialized	Retained
	BAMR0	Initialized	Initialized	Retained
	BBR0	Initialized	Initialized	Retained
	BAR1	Initialized	Initialized	Retained
	BAMR1	Initialized	Initialized	Retained
	BBR1	Initialized	Initialized	Retained
	BAR2	Initialized	Initialized	Retained
	BAMR2	Initialized	Initialized	Retained
	BBR2	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	Register Name	Power-On	Hardware Standby	Sleep
UBC	BAR3	Initialized	Initialized	Retained
	BAMR3	Initialized	Initialized	Retained
	BBR3	Initialized	Initialized	Retained
	BRCR	Initialized	Initialized	Retained
	BAR4	Initialized	Initialized	Retained
	BAMR4	Initialized	Initialized	Retained
BSC	BBR4	Initialized	Initialized	Retained
	BAR5	Initialized	Initialized	Retained
	BAMR5	Initialized	Initialized	Retained
	BBR5	Initialized	Initialized	Retained
	BAR6	Initialized	Initialized	Retained
	BAMR6	Initialized	Initialized	Retained
	BBR6	Initialized	Initialized	Retained
	BAR7	Initialized	Initialized	Retained
	BAMR7	Initialized	Initialized	Retained
	BBR7	Initialized	Initialized	Retained
	BRCR1	Initialized	Initialized	Retained
	CS0BCR	Initialized	Initialized	Retained
	CS1BCR	Initialized	Initialized	Retained
	CS2BCR	Initialized	Initialized	Retained
	CS3BCR	Initialized	Initialized	Retained
	CS0WCR	Initialized	Initialized	Retained
	CS1WCR	Initialized	Initialized	Retained
	CS2WCR	Initialized	Initialized	Retained
	CS3WCR	Initialized	Initialized	Retained
DMAC	SAR0	Initialized	Initialized	Retained
	DAR0	Initialized	Initialized	Retained
	DMATCR0	Initialized	Initialized	Retained
	CHCR0	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	Register Name	Power-On	Hardware Standby	Sleep
DMAC	CHFR0	Initialized	Initialized	Retained
	TEMSK0	Initialized	Initialized	Retained
	RSAR0	Initialized	Initialized	Retained
	RDAR0	Initialized	Initialized	Retained
	RDMATCR0	Initialized	Initialized	Retained
	SAR1	Initialized	Initialized	Retained
	DAR1	Initialized	Initialized	Retained
	DMATCR1	Initialized	Initialized	Retained
	CHCR1	Initialized	Initialized	Retained
	CHFR1	Initialized	Initialized	Retained
	TEMSK1	Initialized	Initialized	Retained
	RSAR1	Initialized	Initialized	Retained
	RDAR1	Initialized	Initialized	Retained
	RDMATCR1	Initialized	Initialized	Retained
	SAR2	Initialized	Initialized	Retained
	DAR2	Initialized	Initialized	Retained
	DMATCR2	Initialized	Initialized	Retained
	CHCR2	Initialized	Initialized	Retained
	CHFR2	Initialized	Initialized	Retained
	TEMSK2	Initialized	Initialized	Retained
	RSAR2	Initialized	Initialized	Retained
	RDAR2	Initialized	Initialized	Retained
	RDMATCR2	Initialized	Initialized	Retained
	SAR3	Initialized	Initialized	Retained
	DAR3	Initialized	Initialized	Retained
	DMATCR3	Initialized	Initialized	Retained
	CHCR3	Initialized	Initialized	Retained
	CHFR3	Initialized	Initialized	Retained
	TEMSK3	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	Register Name	Power-On	Hardware Standby	Sleep
DMAC	RSAR3	Initialized	Initialized	Retained
	RDAR3	Initialized	Initialized	Retained
	RDMATCR3	Initialized	Initialized	Retained
	SAR4	Initialized	Initialized	Retained
	DAR4	Initialized	Initialized	Retained
	DMATCR4	Initialized	Initialized	Retained
	CHCR4	Initialized	Initialized	Retained
	CHFR4	Initialized	Initialized	Retained
	TEMSK4	Initialized	Initialized	Retained
	RSAR4	Initialized	Initialized	Retained
	RDAR4	Initialized	Initialized	Retained
	RDMATCR4	Initialized	Initialized	Retained
	ARCR4	Initialized	Initialized	Retained
	RARCR4	Initialized	Initialized	Retained
	SAR5	Initialized	Initialized	Retained
	DAR5	Initialized	Initialized	Retained
	DMATCR5	Initialized	Initialized	Retained
	CHCR5	Initialized	Initialized	Retained
	CHFR5	Initialized	Initialized	Retained
	TEMSK5	Initialized	Initialized	Retained
	RSAR5	Initialized	Initialized	Retained
	RDAR5	Initialized	Initialized	Retained
	RDMATCR5	Initialized	Initialized	Retained
	ARCR5	Initialized	Initialized	Retained
	RARCR5	Initialized	Initialized	Retained
	SAR6	Initialized	Initialized	Retained
	DAR6	Initialized	Initialized	Retained
	DMATCR6	Initialized	Initialized	Retained
	CHCR6	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	Register Name	Power-On	Hardware Standby	Sleep
DMAC	CHFR6	Initialized	Initialized	Retained
	TEMSK6	Initialized	Initialized	Retained
	RSAR6	Initialized	Initialized	Retained
	RDAR6	Initialized	Initialized	Retained
	RDMATCR6	Initialized	Initialized	Retained
	ARCR6	Initialized	Initialized	Retained
	RARCR6	Initialized	Initialized	Retained
	SAR7	Initialized	Initialized	Retained
	DAR7	Initialized	Initialized	Retained
	DMATCR7	Initialized	Initialized	Retained
	CHCR7	Initialized	Initialized	Retained
A-DMAC	CHFR7	Initialized	Initialized	Retained
	TEMSK7	Initialized	Initialized	Retained
	RSAR7	Initialized	Initialized	Retained
	RDAR7	Initialized	Initialized	Retained
	RDMATCR7	Initialized	Initialized	Retained
	ARCR7	Initialized	Initialized	Retained
	RARCR7	Initialized	Initialized	Retained
	DMAOR	Initialized	Initialized	Retained
	DMAFR	Initialized	Initialized	Retained
	DMARS0	Initialized	Initialized	Retained
	DMARS1	Initialized	Initialized	Retained
	DMARS2	Initialized	Initialized	Retained
	DMARS3	Initialized	Initialized	Retained
	ADMAOR	Initialized	Initialized	Retained
	ADMAABR	Initialized	Initialized	Retained
	ADMAIE0	Initialized	Initialized	Retained
	ADMAIE1	Initialized	Initialized	Retained
	ADMAIE2	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	Register Name	Power-On	Hardware Standby	Sleep
A-DMAC	ADMAIE3	Initialized	Initialized	Retained
	ADMAIE4	Initialized	Initialized	Retained
	ADMAIE5	Initialized	Initialized	Retained
	ADMAIE6	Initialized	Initialized	Retained
	ADMAIE7	Initialized	Initialized	Retained
	ADMAIE8	Initialized	Initialized	Retained
	ADMAIE9	Initialized	Initialized	Retained
	ADMADV0	Initialized	Initialized	Retained
	ADMADV1	Initialized	Initialized	Retained
	ADMADV2	Initialized	Initialized	Retained
	ADMADV3	Initialized	Initialized	Retained
	ADMADV4	Initialized	Initialized	Retained
	ADMADV5	Initialized	Initialized	Retained
	ADMADV6	Initialized	Initialized	Retained
	ADMATE0	Initialized	Initialized	Retained
	ADMATE1	Initialized	Initialized	Retained
	ADMATE2	Initialized	Initialized	Retained
	ADMADE0	Initialized	Initialized	Retained
	ADMADE1	Initialized	Initialized	Retained
	ADMADE2	Initialized	Initialized	Retained
	ADMADE3	Initialized	Initialized	Retained
	ADMADE4	Initialized	Initialized	Retained
	ADMADE5	Initialized	Initialized	Retained
	ADMADE6	Initialized	Initialized	Retained
	ADMADE7	Initialized	Initialized	Retained
	ADMAMODE0	Initialized	Initialized	Retained
	ADMAMODE1	Initialized	Initialized	Retained
	ADMAMODE2	Initialized	Initialized	Retained
	ADMAMODE3	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	Register Name	Power-On	Hardware Standby	Sleep
A-DMAC	ADMATCR0	Initialized	Initialized	Retained
	ADMARTCR0	Initialized	Initialized	Retained
	ADMATCR1	Initialized	Initialized	Retained
	ADMARTCR1	Initialized	Initialized	Retained
	ADMATCR56	Initialized	Initialized	Retained
	ADMATCR57	Initialized	Initialized	Retained
	ADMATCR58	Initialized	Initialized	Retained
	ADMATCR59	Initialized	Initialized	Retained
	ADMATCR60	Initialized	Initialized	Retained
	ADMATCR61	Initialized	Initialized	Retained
	ADMATCR62	Initialized	Initialized	Retained
	ADMATCR63	Initialized	Initialized	Retained
	ADMATCR64	Initialized	Initialized	Retained
	ADMATCR65	Initialized	Initialized	Retained
	ADMATCR66	Initialized	Initialized	Retained
	ADMATCR67	Initialized	Initialized	Retained
	ADMATCR68	Initialized	Initialized	Retained
	ADMATCR69	Initialized	Initialized	Retained
	ADMATCR70	Initialized	Initialized	Retained
	ADMATCR71	Initialized	Initialized	Retained
	ADMAAR0	Initialized	Initialized	Retained
	ADMARAR0	Initialized	Initialized	Retained
	ADMAAR1	Initialized	Initialized	Retained
	ADMARAR1	Initialized	Initialized	Retained
	ADMAAR56	Initialized	Initialized	Retained
	ADMAAR57	Initialized	Initialized	Retained
	ADMAAR58	Initialized	Initialized	Retained
	ADMAAR59	Initialized	Initialized	Retained
	ADMAAR60	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	Register Name	Power-On	Hardware Standby	Sleep
A-DMAC	ADMAAR61	Initialized	Initialized	Retained
	ADMAAR62	Initialized	Initialized	Retained
	ADMAAR63	Initialized	Initialized	Retained
	ADMAAR64	Initialized	Initialized	Retained
	ADMAAR65	Initialized	Initialized	Retained
	ADMAAR66	Initialized	Initialized	Retained
	ADMAAR67	Initialized	Initialized	Retained
	ADMAAR68	Initialized	Initialized	Retained
	ADMAAR69	Initialized	Initialized	Retained
	ADMAAR70	Initialized	Initialized	Retained
	ADMAAR71	Initialized	Initialized	Retained
	ADMABUF2	Initialized	Initialized	Retained
	ADMABUF3	Initialized	Initialized	Retained
	ADMABUF4	Initialized	Initialized	Retained
	ADMABUF5	Initialized	Initialized	Retained
	ADMABUF6	Initialized	Initialized	Retained
	ADMABUF7	Initialized	Initialized	Retained
	ADMARVPR0	Initialized	Initialized	Retained
	ADMARVPR1	Initialized	Initialized	Retained
	ADMARVPR2	Initialized	Initialized	Retained
	ADMARVPR3	Initialized	Initialized	Retained
	ADMARVPR4	Initialized	Initialized	Retained
	ADMARVPR5	Initialized	Initialized	Retained
	ADMARVPR6	Initialized	Initialized	Retained
	ADMARVPR7	Initialized	Initialized	Retained
	ADMATVPR0	Initialized	Initialized	Retained
	ADMATVPR1	Initialized	Initialized	Retained
	ADMATVPR2	Initialized	Initialized	Retained
	ADMATVPR3	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	Register Name	Power-On	Hardware Standby	Sleep
A-DMAC	ADMATVPR4	Initialized	Initialized	Retained
	ADMATVPR5	Initialized	Initialized	Retained
	ADMATVPR6	Initialized	Initialized	Retained
	ADMATVPR7	Initialized	Initialized	Retained
	ADMAFRWR0	Initialized	Initialized	Retained
	ADMAFRWR1	Initialized	Initialized	Retained
	ADMAFRWR2	Initialized	Initialized	Retained
	ADMAFRWR3	Initialized	Initialized	Retained
	ADMAFRLMB	Initialized	Initialized	Retained
	ADMAFRGENCTR	Initialized	Initialized	Retained
	ADMAFRCTR	Initialized	Initialized	Retained
	ADMAFRFCTR	Initialized	Initialized	Retained
	ADMAFRTRSTAT	Initialized	Initialized	Retained
	ADMAFRSTAT	Initialized	Initialized	Retained
	ADMAFRFSTAT	Initialized	Initialized	Retained
ATU-III	ATUENR	Initialized	Initialized	Retained
	CBCNT	Initialized	Initialized	Retained
	NCMR	Initialized	Initialized	Retained
	PSCR0	Initialized	Initialized	Retained
	PSCR1	Initialized	Initialized	Retained
	PSCR2	Initialized	Initialized	Retained
	PSCR3	Initialized	Initialized	Retained
	TCRA	Initialized	Initialized	Retained
	TIOR1A	Initialized	Initialized	Retained
	TIOR2A	Initialized	Initialized	Retained
	TSRA	Initialized	Initialized	Retained
	TIERA	Initialized	Initialized	Retained
	NCMCR1A	Initialized	Initialized	Retained
	NCNTA0	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	Register Name	Power-On	Hardware Standby	Sleep
ATU-III	NCRA0	Initialized	Initialized	Retained
	NCNTA1	Initialized	Initialized	Retained
	NCRA1	Initialized	Initialized	Retained
	NCNTA2	Initialized	Initialized	Retained
	NCRA2	Initialized	Initialized	Retained
	NCNTA3	Initialized	Initialized	Retained
	NCRA3	Initialized	Initialized	Retained
	NCNTA4	Initialized	Initialized	Retained
	NCRA4	Initialized	Initialized	Retained
	NCNTA5	Initialized	Initialized	Retained
	NCRA5	Initialized	Initialized	Retained
	TCNTA	Initialized	Initialized	Retained
	ICRA0	Initialized	Initialized	Retained
	ICRA1	Initialized	Initialized	Retained
	ICRA2	Initialized	Initialized	Retained
	ICRA3	Initialized	Initialized	Retained
	ICRA4	Initialized	Initialized	Retained
	ICRA5	Initialized	Initialized	Retained
	TCRB	Initialized	Initialized	Retained
	TIORB	Initialized	Initialized	Retained
	TSRB	Initialized	Initialized	Retained
	TIERB	Initialized	Initialized	Retained
	TCNTB0	Initialized	Initialized	Retained
	ICRB0	Initialized	Initialized	Retained
	OCRB0	Initialized	Initialized	Retained
	TCNTB1	Initialized	Initialized	Retained
	OCRB1	Initialized	Initialized	Retained
	ICRB1	Initialized	Initialized	Retained
	ICRB2	Initialized	Initialized	Retained
	LDB	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	Register Name	Power-On	Hardware Standby	Sleep
ATU-III	RLDB	Initialized	Initialized	Retained
	PIMR	Initialized	Initialized	Retained
	TCNTB2	Initialized	Initialized	Retained
	TCNTB6	Initialized	Initialized	Retained
	OCRB6	Initialized	Initialized	Retained
	OCRB7	Initialized	Initialized	Retained
	TCNTB3	Initialized	Initialized	Retained
	TCNTB4	Initialized	Initialized	Retained
	TCNTB5	Initialized	Initialized	Retained
	TCCLRB	Initialized	Initialized	Retained
	TSTRC	Initialized	Initialized	Retained
	NCCRC0	Initialized	Initialized	Retained
	NCCRC1	Initialized	Initialized	Retained
	NCCRC2	Initialized	Initialized	Retained
	NCCRC3	Initialized	Initialized	Retained
	NCCRC4	Initialized	Initialized	Retained
	NCNTC00	Initialized	Initialized	Retained
	NCNTC01	Initialized	Initialized	Retained
	NCNTC02	Initialized	Initialized	Retained
	NCNTC03	Initialized	Initialized	Retained
	NCRC00	Initialized	Initialized	Retained
	NCRC01	Initialized	Initialized	Retained
	NCRC02	Initialized	Initialized	Retained
	NCRC03	Initialized	Initialized	Retained
	NCNTC10	Initialized	Initialized	Retained
	NCNTC11	Initialized	Initialized	Retained
	NCNTC12	Initialized	Initialized	Retained
	NCNTC13	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	Register Name	Power-On	Hardware Standby	Sleep
ATU-III	NCRC10	Initialized	Initialized	Retained
	NCRC11	Initialized	Initialized	Retained
	NCRC12	Initialized	Initialized	Retained
	NCRC13	Initialized	Initialized	Retained
	NCNTC20	Initialized	Initialized	Retained
	NCNTC21	Initialized	Initialized	Retained
	NCNTC22	Initialized	Initialized	Retained
	NCNTC23	Initialized	Initialized	Retained
	NCRC20	Initialized	Initialized	Retained
	NCRC21	Initialized	Initialized	Retained
	NCRC22	Initialized	Initialized	Retained
	NCRC23	Initialized	Initialized	Retained
	NCNTC30	Initialized	Initialized	Retained
	NCNTC31	Initialized	Initialized	Retained
	NCNTC32	Initialized	Initialized	Retained
	NCNTC33	Initialized	Initialized	Retained
	NCRC30	Initialized	Initialized	Retained
	NCRC31	Initialized	Initialized	Retained
	NCRC32	Initialized	Initialized	Retained
	NCRC33	Initialized	Initialized	Retained
	NCNTC40	Initialized	Initialized	Retained
	NCNTC41	Initialized	Initialized	Retained
	NCNTC42	Initialized	Initialized	Retained
	NCNTC43	Initialized	Initialized	Retained
	NCRC40	Initialized	Initialized	Retained
	NCRC41	Initialized	Initialized	Retained
	NCRC42	Initialized	Initialized	Retained
	NCRC43	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	Register Name	Power-On	Hardware Standby	Sleep
ATU-III	TCRC0	Initialized	Initialized	Retained
	TIERC0	Initialized	Initialized	Retained
	TIORC0	Initialized	Initialized	Retained
	TSRC0	Initialized	Initialized	Retained
	GRC00	Initialized	Initialized	Retained
	GRC01	Initialized	Initialized	Retained
	GRC02	Initialized	Initialized	Retained
	GRC03	Initialized	Initialized	Retained
	TCNTC0	Initialized	Initialized	Retained
	TCRC1	Initialized	Initialized	Retained
	TIERC1	Initialized	Initialized	Retained
	TIORC1	Initialized	Initialized	Retained
	TSRC1	Initialized	Initialized	Retained
	GRC10	Initialized	Initialized	Retained
	GRC11	Initialized	Initialized	Retained
	GRC12	Initialized	Initialized	Retained
	GRC13	Initialized	Initialized	Retained
	TCNTC1	Initialized	Initialized	Retained
	TCRC2	Initialized	Initialized	Retained
	TIERC2	Initialized	Initialized	Retained
	TIORC2	Initialized	Initialized	Retained
	TSRC2	Initialized	Initialized	Retained
	GRC20	Initialized	Initialized	Retained
	GRC21	Initialized	Initialized	Retained
	GRC22	Initialized	Initialized	Retained
	GRC23	Initialized	Initialized	Retained
	TCNTC2	Initialized	Initialized	Retained
	TCRC3	Initialized	Initialized	Retained
	TIERC3	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	**Register Name**	**Power-On**	**Hardware Standby**	**Sleep**
ATU-III	TIORC3	Initialized	Initialized	Retained
	TSRC3	Initialized	Initialized	Retained
	GRC30	Initialized	Initialized	Retained
	GRC31	Initialized	Initialized	Retained
	GRC32	Initialized	Initialized	Retained
	GRC33	Initialized	Initialized	Retained
	TCNTC3	Initialized	Initialized	Retained
	TCRC4	Initialized	Initialized	Retained
	TIERC4	Initialized	Initialized	Retained
	TIORC4	Initialized	Initialized	Retained
	TSRC4	Initialized	Initialized	Retained
	GRC40	Initialized	Initialized	Retained
	GRC41	Initialized	Initialized	Retained
	GRC42	Initialized	Initialized	Retained
	GRC43	Initialized	Initialized	Retained
	TCNTC4	Initialized	Initialized	Retained
	TSTRD	Initialized	Initialized	Retained
	TCNT1D0	Initialized	Initialized	Retained
	TCNT2D0	Initialized	Initialized	Retained
	OSBRD0	Initialized	Initialized	Retained
	TCRD0	Initialized	Initialized	Retained
	TOCRD0	Initialized	Initialized	Retained
	CMPOD0	Initialized	Initialized	Retained
	TCNT1D1	Initialized	Initialized	Retained
	TCNT2D1	Initialized	Initialized	Retained
	OSBRD1	Initialized	Initialized	Retained
	TCRD1	Initialized	Initialized	Retained
	TOCRD1	Initialized	Initialized	Retained
	CMPOD1	Initialized	Initialized	Retained

Module	Register Name	Reset State	Power-Down State	
		Power-On	Hardware Standby	Sleep
ATU-III	TCNT1D2	Initialized	Initialized	Retained
	TCNT2D2	Initialized	Initialized	Retained
	OSBRD2	Initialized	Initialized	Retained
	TCRD2	Initialized	Initialized	Retained
	TOCRD2	Initialized	Initialized	Retained
	TCNT1D3	Initialized	Initialized	Retained
	TCNT2D3	Initialized	Initialized	Retained
	OSBRD3	Initialized	Initialized	Retained
	TCRD3	Initialized	Initialized	Retained
	TOCRD3	Initialized	Initialized	Retained
	TIOR1D0	Initialized	Initialized	Retained
	TIOR2D0	Initialized	Initialized	Retained
	DSTRD0	Initialized	Initialized	Retained
	DSRD0	Initialized	Initialized	Retained
	DCRD0	Initialized	Initialized	Retained
	TSRD0	Initialized	Initialized	Retained
	TIERD0	Initialized	Initialized	Retained
	OCRD00	Initialized	Initialized	Retained
	OCRD01	Initialized	Initialized	Retained
	OCRD02	Initialized	Initialized	Retained
	OCRD03	Initialized	Initialized	Retained
	GRD00	Initialized	Initialized	Retained
	GRD01	Initialized	Initialized	Retained
	GRD02	Initialized	Initialized	Retained
	GRD03	Initialized	Initialized	Retained
	DCNTD00	Initialized	Initialized	Retained
	DCNTD01	Initialized	Initialized	Retained
	DCNTD02	Initialized	Initialized	Retained
	DCNTD03	Initialized	Initialized	Retained
	TIOR1D1	Initialized	Initialized	Retained
	TIOR2D1	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	**Register Name**	**Power-On**	**Hardware Standby**	**Sleep**
ATU-III	DSTRD1	Initialized	Initialized	Retained
	DSRD1	Initialized	Initialized	Retained
	DCRD1	Initialized	Initialized	Retained
	TSRD1	Initialized	Initialized	Retained
	TIERD1	Initialized	Initialized	Retained
	OCRD10	Initialized	Initialized	Retained
	OCRD11	Initialized	Initialized	Retained
	OCRD12	Initialized	Initialized	Retained
	OCRD13	Initialized	Initialized	Retained
	GRD10	Initialized	Initialized	Retained
	GRD11	Initialized	Initialized	Retained
	GRD12	Initialized	Initialized	Retained
	GRD13	Initialized	Initialized	Retained
	DCNTD10	Initialized	Initialized	Retained
	DCNTD11	Initialized	Initialized	Retained
	DCNTD12	Initialized	Initialized	Retained
	DCNTD13	Initialized	Initialized	Retained
	TIOR1D2	Initialized	Initialized	Retained
	TIOR2D2	Initialized	Initialized	Retained
	DSTRD2	Initialized	Initialized	Retained
	DSRD2	Initialized	Initialized	Retained
	DCRD2	Initialized	Initialized	Retained
	TSRD2	Initialized	Initialized	Retained
	TIERD2	Initialized	Initialized	Retained
	OCRD20	Initialized	Initialized	Retained
	OCRD21	Initialized	Initialized	Retained
	OCRD22	Initialized	Initialized	Retained
	OCRD23	Initialized	Initialized	Retained
	GRD20	Initialized	Initialized	Retained
	GRD21	Initialized	Initialized	Retained
	GRD22	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	Register Name	Power-On	Hardware Standby	Sleep
ATU-III	GRD23	Initialized	Initialized	Retained
	DCNTD20	Initialized	Initialized	Retained
	DCNTD21	Initialized	Initialized	Retained
	DCNTD22	Initialized	Initialized	Retained
	DCNTD23	Initialized	Initialized	Retained
	TIOR1D3	Initialized	Initialized	Retained
	TIOR2D3	Initialized	Initialized	Retained
	DSTRD3	Initialized	Initialized	Retained
	DSRD3	Initialized	Initialized	Retained
	DCRD3	Initialized	Initialized	Retained
	TSRD3	Initialized	Initialized	Retained
	TIERD3	Initialized	Initialized	Retained
	OCRD30	Initialized	Initialized	Retained
	OCRD31	Initialized	Initialized	Retained
	OCRD32	Initialized	Initialized	Retained
	OCRD33	Initialized	Initialized	Retained
	GRD30	Initialized	Initialized	Retained
	GRD31	Initialized	Initialized	Retained
	GRD32	Initialized	Initialized	Retained
	GRD33	Initialized	Initialized	Retained
	DCNTD30	Initialized	Initialized	Retained
	DCNTD31	Initialized	Initialized	Retained
	DCNTD32	Initialized	Initialized	Retained
	DCNTD33	Initialized	Initialized	Retained
	TSTRE	Initialized	Initialized	Retained
	TCRE0	Initialized	Initialized	Retained
	TOCRE0	Initialized	Initialized	Retained
	TIERE0	Initialized	Initialized	Retained
	RLDCRE0	Initialized	Initialized	Retained
	TSRE0	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	Register Name	Power-On	Hardware Standby	Sleep
ATU-III	PSCRE0	Initialized	Initialized	Retained
	SSTRE0	Initialized	Initialized	Retained
	CYLRE00	Initialized	Initialized	Retained
	CYLRE01	Initialized	Initialized	Retained
	CYLRE02	Initialized	Initialized	Retained
	CYLRE03	Initialized	Initialized	Retained
	DTRE00	Initialized	Initialized	Retained
	DTRE01	Initialized	Initialized	Retained
	DTRE02	Initialized	Initialized	Retained
	DTRE03	Initialized	Initialized	Retained
	CRLDE00	Initialized	Initialized	Retained
	CRLDE01	Initialized	Initialized	Retained
	CRLDE02	Initialized	Initialized	Retained
	DRLDE00	Initialized	Initialized	Retained
	DRLDE01	Initialized	Initialized	Retained
	DRLDE02	Initialized	Initialized	Retained
	DRLDE03	Initialized	Initialized	Retained
	TCNTE00	Initialized	Initialized	Retained
	TCNTE01	Initialized	Initialized	Retained
	TCNTE02	Initialized	Initialized	Retained
	TCNTE03	Initialized	Initialized	Retained
	PSCCRE00	Initialized	Initialized	Retained
	PSCCRE01	Initialized	Initialized	Retained
	PSCCRE02	Initialized	Initialized	Retained
	PSCCRE03	Initialized	Initialized	Retained
	TCRE1	Initialized	Initialized	Retained
	TOCRE1	Initialized	Initialized	Retained
	TIERE1	Initialized	Initialized	Retained
	RLDCRE1	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	Register Name	Power-On	Hardware Standby	Sleep
ATU-III	TSRE1	Initialized	Initialized	Retained
	PSCRE1	Initialized	Initialized	Retained
	SSTRE1	Initialized	Initialized	Retained
	CYLRE10	Initialized	Initialized	Retained
	CYLRE11	Initialized	Initialized	Retained
	CYLRE12	Initialized	Initialized	Retained
	CYLRE13	Initialized	Initialized	Retained
	DTRE10	Initialized	Initialized	Retained
	DTRE11	Initialized	Initialized	Retained
	DTRE12	Initialized	Initialized	Retained
	DTRE13	Initialized	Initialized	Retained
	CRLDE10	Initialized	Initialized	Retained
	CRLDE11	Initialized	Initialized	Retained
	CRLDE12	Initialized	Initialized	Retained
	CRLDE13	Initialized	Initialized	Retained
	DRLDE10	Initialized	Initialized	Retained
	DRLDE11	Initialized	Initialized	Retained
	DRLDE12	Initialized	Initialized	Retained
	DRLDE13	Initialized	Initialized	Retained
	TCNTE10	Initialized	Initialized	Retained
	TCNTE11	Initialized	Initialized	Retained
	TCNTE12	Initialized	Initialized	Retained
	TCNTE13	Initialized	Initialized	Retained
	PSCCRE10	Initialized	Initialized	Retained
	PSCCRE11	Initialized	Initialized	Retained
	PSCCRE12	Initialized	Initialized	Retained
	PSCCRE13	Initialized	Initialized	Retained
	TCRE2	Initialized	Initialized	Retained
	TOCRE2	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	**Register Name**	**Power-On**	**Hardware Standby**	**Sleep**
ATU-III	TIERE2	Initialized	Initialized	Retained
	RLDCRE2	Initialized	Initialized	Retained
	TSRE2	Initialized	Initialized	Retained
	PSCRE2	Initialized	Initialized	Retained
	SSTRE2	Initialized	Initialized	Retained
	CYLRE20	Initialized	Initialized	Retained
	CYLRE21	Initialized	Initialized	Retained
	CYLRE22	Initialized	Initialized	Retained
	CYLRE23	Initialized	Initialized	Retained
	DTRE20	Initialized	Initialized	Retained
	DTRE21	Initialized	Initialized	Retained
	DTRE22	Initialized	Initialized	Retained
	DTRE23	Initialized	Initialized	Retained
	CRLDE20	Initialized	Initialized	Retained
	CRLDE21	Initialized	Initialized	Retained
	CRLDE22	Initialized	Initialized	Retained
	CRLDE23	Initialized	Initialized	Retained
	DRLDE20	Initialized	Initialized	Retained
	DRLDE21	Initialized	Initialized	Retained
	DRLDE22	Initialized	Initialized	Retained
	DRLDE23	Initialized	Initialized	Retained
	TCNTE20	Initialized	Initialized	Retained
	TCNTE21	Initialized	Initialized	Retained
	TCNTE22	Initialized	Initialized	Retained
	TCNTE23	Initialized	Initialized	Retained
	PSCCRE20	Initialized	Initialized	Retained
	PSCCRE21	Initialized	Initialized	Retained
	PSCCRE22	Initialized	Initialized	Retained
	PSCCRE23	Initialized	Initialized	Retained
	TCRE3	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	Register Name	Power-On	Hardware Standby	Sleep
ATU-III	TOCRE3	Initialized	Initialized	Retained
	TIERE3	Initialized	Initialized	Retained
	RLDCRE3	Initialized	Initialized	Retained
	TSRE3	Initialized	Initialized	Retained
	PSCRE3	Initialized	Initialized	Retained
	SSTRE3	Initialized	Initialized	Retained
	CYLRE30	Initialized	Initialized	Retained
	CYLRE31	Initialized	Initialized	Retained
	CYLRE32	Initialized	Initialized	Retained
	CYLRE33	Initialized	Initialized	Retained
	DTRE30	Initialized	Initialized	Retained
	DTRE31	Initialized	Initialized	Retained
	DTRE32	Initialized	Initialized	Retained
	DTRE33	Initialized	Initialized	Retained
	CRLDE30	Initialized	Initialized	Retained
	CRLDE31	Initialized	Initialized	Retained
	CRLDE32	Initialized	Initialized	Retained
	CRLDE33	Initialized	Initialized	Retained
	DRLDE30	Initialized	Initialized	Retained
	DRLDE31	Initialized	Initialized	Retained
	DRLDE32	Initialized	Initialized	Retained
	DRLDE33	Initialized	Initialized	Retained
	TCNTE30	Initialized	Initialized	Retained
	TCNTE31	Initialized	Initialized	Retained
	TCNTE32	Initialized	Initialized	Retained
	TCNTE33	Initialized	Initialized	Retained
	PSCCRE30	Initialized	Initialized	Retained
	PSCCRE31	Initialized	Initialized	Retained
	PSCCRE32	Initialized	Initialized	Retained
	PSCCRE33	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	Register Name	Power-On	Hardware Standby	Sleep
ATU-III	TCRE4	Initialized	Initialized	Retained
	TOCRE4	Initialized	Initialized	Retained
	TIERE4	Initialized	Initialized	Retained
	RLDCRE4	Initialized	Initialized	Retained
	TSRE4	Initialized	Initialized	Retained
	PSCRE4	Initialized	Initialized	Retained
	SSTRE4	Initialized	Initialized	Retained
	CYLRE40	Initialized	Initialized	Retained
	CYLRE41	Initialized	Initialized	Retained
	CYLRE42	Initialized	Initialized	Retained
	CYLRE43	Initialized	Initialized	Retained
	DTRE40	Initialized	Initialized	Retained
	DTRE41	Initialized	Initialized	Retained
	DTRE42	Initialized	Initialized	Retained
	DTRE43	Initialized	Initialized	Retained
	CRLDE40	Initialized	Initialized	Retained
	CRLDE41	Initialized	Initialized	Retained
	CRLDE42	Initialized	Initialized	Retained
	CRLDE43	Initialized	Initialized	Retained
	DRLDE40	Initialized	Initialized	Retained
	DRLDE41	Initialized	Initialized	Retained
	DRLDE42	Initialized	Initialized	Retained
	DRLDE43	Initialized	Initialized	Retained
	TCNTE40	Initialized	Initialized	Retained
	TCNTE41	Initialized	Initialized	Retained
	TCNTE42	Initialized	Initialized	Retained
	TCNTE43	Initialized	Initialized	Retained
	PSCCRE40	Initialized	Initialized	Retained
	PSCCRE41	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	Register Name	Power-On	Hardware Standby	Sleep
ATU-III	PSCCRE42	Initialized	Initialized	Retained
	PSCCRE43	Initialized	Initialized	Retained
	TCRE5	Initialized	Initialized	Retained
	TOCRE5	Initialized	Initialized	Retained
	TIERE5	Initialized	Initialized	Retained
	RLDCRE5	Initialized	Initialized	Retained
	TSRE5	Initialized	Initialized	Retained
	PSCRE5	Initialized	Initialized	Retained
	SSTRE5	Initialized	Initialized	Retained
	CYLRE50	Initialized	Initialized	Retained
	CYLRE51	Initialized	Initialized	Retained
	CYLRE52	Initialized	Initialized	Retained
	CYLRE53	Initialized	Initialized	Retained
	DTRE50	Initialized	Initialized	Retained
	DTRE51	Initialized	Initialized	Retained
	DTRE52	Initialized	Initialized	Retained
	DTRE53	Initialized	Initialized	Retained
	CRLDE50	Initialized	Initialized	Retained
	CRLDE51	Initialized	Initialized	Retained
	CRLDE52	Initialized	Initialized	Retained
	CRLDE53	Initialized	Initialized	Retained
	DRLDE50	Initialized	Initialized	Retained
	DRLDE51	Initialized	Initialized	Retained
	DRLDE52	Initialized	Initialized	Retained
	DRLDE53	Initialized	Initialized	Retained
	TCNTE50	Initialized	Initialized	Retained
	TCNTE51	Initialized	Initialized	Retained
	TCNTE52	Initialized	Initialized	Retained
	TCNTE53	Initialized	Initialized	Retained
	PSCCRE50	Initialized	Initialized	Retained
	PSCCRE51	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	Register Name	Power-On	Hardware Standby	Sleep
ATU-III	PSCCRE52	Initialized	Initialized	Retained
	PSCCRE53	Initialized	Initialized	Retained
	TCRE6	Initialized	Initialized	Retained
	TOCRE6	Initialized	Initialized	Retained
	TIERE6	Initialized	Initialized	Retained
	RLDCRE6	Initialized	Initialized	Retained
	TSRE6	Initialized	Initialized	Retained
	PSCRE6	Initialized	Initialized	Retained
	SSTRE6	Initialized	Initialized	Retained
	CYLRE60	Initialized	Initialized	Retained
	CYLRE61	Initialized	Initialized	Retained
	CYLRE62	Initialized	Initialized	Retained
	CYLRE63	Initialized	Initialized	Retained
	DTRE60	Initialized	Initialized	Retained
	DTRE61	Initialized	Initialized	Retained
	DTRE62	Initialized	Initialized	Retained
	DTRE63	Initialized	Initialized	Retained
	CRLDE60	Initialized	Initialized	Retained
	CRLDE61	Initialized	Initialized	Retained
	CRLDE62	Initialized	Initialized	Retained
	CRLDE63	Initialized	Initialized	Retained
	DRLDE60	Initialized	Initialized	Retained
	DRLDE61	Initialized	Initialized	Retained
	DRLDE62	Initialized	Initialized	Retained
	DRLDE63	Initialized	Initialized	Retained
	TCNTE60	Initialized	Initialized	Retained
	TCNTE61	Initialized	Initialized	Retained
	TCNTE62	Initialized	Initialized	Retained
	TCNTE63	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	Register Name	Power-On	Hardware Standby	Sleep
ATU-III	PSCCRE60	Initialized	Initialized	Retained
	PSCCRE61	Initialized	Initialized	Retained
	PSCCRE62	Initialized	Initialized	Retained
	PSCCRE63	Initialized	Initialized	Retained
	TSTRF	Initialized	Initialized	Retained
	NCCRF	Initialized	Initialized	Retained
	NCMCR1F	Initialized	Initialized	Retained
	NCNTFA0	Initialized	Initialized	Retained
	NCRFA0	Initialized	Initialized	Retained
	NCNTFA1	Initialized	Initialized	Retained
	NCRFA1	Initialized	Initialized	Retained
	NCNTFA2	Initialized	Initialized	Retained
	NCRFA2	Initialized	Initialized	Retained
	NCNTFA3	Initialized	Initialized	Retained
	NCRFA3	Initialized	Initialized	Retained
	NCNTFA4	Initialized	Initialized	Retained
	NCRFA4	Initialized	Initialized	Retained
	NCNTFA5	Initialized	Initialized	Retained
	NCRFA5	Initialized	Initialized	Retained
	NCNTFA6	Initialized	Initialized	Retained
	NCRFA6	Initialized	Initialized	Retained
	NCNTFA7	Initialized	Initialized	Retained
	NCRFA7	Initialized	Initialized	Retained
	NCNTFA8	Initialized	Initialized	Retained
	NCRFA8	Initialized	Initialized	Retained
	NCNTFA9	Initialized	Initialized	Retained
	NCRFA9	Initialized	Initialized	Retained
	NCNTFA10	Initialized	Initialized	Retained
	NCRFA10	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	**Register Name**	**Power-On**	**Hardware Standby**	**Sleep**
ATU-III	NCNTFA11	Initialized	Initialized	Retained
	NCRFA11	Initialized	Initialized	Retained
	NCNTFA12	Initialized	Initialized	Retained
	NCRFA12	Initialized	Initialized	Retained
	NCNTFA13	Initialized	Initialized	Retained
	NCRFA13	Initialized	Initialized	Retained
	NCNTFA14	Initialized	Initialized	Retained
	NCRFA14	Initialized	Initialized	Retained
	NCNTFA15	Initialized	Initialized	Retained
	NCRFA15	Initialized	Initialized	Retained
	NCNTFA16	Initialized	Initialized	Retained
	NCRFA16	Initialized	Initialized	Retained
	NCNTFA17	Initialized	Initialized	Retained
	NCRFA17	Initialized	Initialized	Retained
	NCNTFA18	Initialized	Initialized	Retained
	NCRFA18	Initialized	Initialized	Retained
	NCNTFA19	Initialized	Initialized	Retained
	NCRFA19	Initialized	Initialized	Retained
	NCNTFA20	Initialized	Initialized	Retained
	NCRFA20	Initialized	Initialized	Retained
	NCNTFA21	Initialized	Initialized	Retained
	NCRFA21	Initialized	Initialized	Retained
	NCNTFA22	Initialized	Initialized	Retained
	NCRFA22	Initialized	Initialized	Retained
	NCNTFA23	Initialized	Initialized	Retained
	NCRFA23	Initialized	Initialized	Retained
	NCNTFA24	Initialized	Initialized	Retained
	NCRFA24	Initialized	Initialized	Retained
	NCNTFA25	Initialized	Initialized	Retained
	NCRFA25	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	Register Name	Power-On	Hardware Standby	Sleep
ATU-III	NCNTFA26	Initialized	Initialized	Retained
	NCRFA26	Initialized	Initialized	Retained
	NCNTFA27	Initialized	Initialized	Retained
	NCRFA27	Initialized	Initialized	Retained
	NCNTFB0	Initialized	Initialized	Retained
	NCRFB0	Initialized	Initialized	Retained
	NCNTFB1	Initialized	Initialized	Retained
	NCRFB1	Initialized	Initialized	Retained
	NCNTFB2	Initialized	Initialized	Retained
	NCRFB2	Initialized	Initialized	Retained
	TCRF0	Initialized	Initialized	Retained
	TIERF0	Initialized	Initialized	Retained
	TSRF0	Initialized	Initialized	Retained
	ECNTAF0	Initialized	Initialized	Retained
	ECNTBF0	Initialized	Initialized	Retained
	GRBF0	Initialized	Initialized	Retained
	ECNTCF0	Initialized	Initialized	Retained
	GRAF0	Initialized	Initialized	Retained
	CDRF0	Initialized	Initialized	Retained
	GRCF0	Initialized	Initialized	Retained
	TCRF1	Initialized	Initialized	Retained
	TIERF1	Initialized	Initialized	Retained
	TSRF1	Initialized	Initialized	Retained
	ECNTAF1	Initialized	Initialized	Retained
	ECNTBF1	Initialized	Initialized	Retained
	GRBF1	Initialized	Initialized	Retained
	ECNTCF1	Initialized	Initialized	Retained
	GRAF1	Initialized	Initialized	Retained
	CDRF1	Initialized	Initialized	Retained
	GRCF1	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	Register Name	Power-On	Hardware Standby	Sleep
ATU-III	TCRF2	Initialized	Initialized	Retained
	TIERF2	Initialized	Initialized	Retained
	TSRF2	Initialized	Initialized	Retained
	ECNTAF2	Initialized	Initialized	Retained
	ECNTBF2	Initialized	Initialized	Retained
	GRBF2	Initialized	Initialized	Retained
	ECNTCF2	Initialized	Initialized	Retained
	GRAF2	Initialized	Initialized	Retained
	CDRF2	Initialized	Initialized	Retained
	GRCF2	Initialized	Initialized	Retained
	TCRF3	Initialized	Initialized	Retained
	TIERF3	Initialized	Initialized	Retained
	TSRF3	Initialized	Initialized	Retained
	ECNTAF3	Initialized	Initialized	Retained
	ECNTBF3	Initialized	Initialized	Retained
	GRBF3	Initialized	Initialized	Retained
	ECNTCF3	Initialized	Initialized	Retained
	GRAF3	Initialized	Initialized	Retained
	CDRF3	Initialized	Initialized	Retained
	GRCF3	Initialized	Initialized	Retained
	TCRF4	Initialized	Initialized	Retained
	TIERF4	Initialized	Initialized	Retained
	TSRF4	Initialized	Initialized	Retained
	ECNTAF4	Initialized	Initialized	Retained
	ECNTBF4	Initialized	Initialized	Retained
	GRBF4	Initialized	Initialized	Retained
	ECNTCF4	Initialized	Initialized	Retained
	GRAF4	Initialized	Initialized	Retained
	CDRF4	Initialized	Initialized	Retained
	GRCF4	Initialized	Initialized	Retained
	TCRF5	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	Register Name	Power-On	Hardware Standby	Sleep
ATU-III	TIERF5	Initialized	Initialized	Retained
	TSRF5	Initialized	Initialized	Retained
	ECNTAF5	Initialized	Initialized	Retained
	ECNTBF5	Initialized	Initialized	Retained
	GRBF5	Initialized	Initialized	Retained
	ECNTCF5	Initialized	Initialized	Retained
	GRAF5	Initialized	Initialized	Retained
	CDRF5	Initialized	Initialized	Retained
	GRCF5	Initialized	Initialized	Retained
	TCRF6	Initialized	Initialized	Retained
	TIERF6	Initialized	Initialized	Retained
	TSRF6	Initialized	Initialized	Retained
	ECNTAF6	Initialized	Initialized	Retained
	ECNTBF6	Initialized	Initialized	Retained
	GRBF6	Initialized	Initialized	Retained
	ECNTCF6	Initialized	Initialized	Retained
	GRAF6	Initialized	Initialized	Retained
	CDRF6	Initialized	Initialized	Retained
	GRCF6	Initialized	Initialized	Retained
	TCRF7	Initialized	Initialized	Retained
	TIERF7	Initialized	Initialized	Retained
	TSRF7	Initialized	Initialized	Retained
	ECNTAF7	Initialized	Initialized	Retained
	ECNTBF7	Initialized	Initialized	Retained
	GRBF7	Initialized	Initialized	Retained
	ECNTCF7	Initialized	Initialized	Retained
	GRAF7	Initialized	Initialized	Retained
	CDRF7	Initialized	Initialized	Retained
	GRCF7	Initialized	Initialized	Retained
	TCRF8	Initialized	Initialized	Retained
	TIERF8	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	Register Name	Power-On	Hardware Standby	Sleep
ATU-III	TSRF8	Initialized	Initialized	Retained
	ECNTAF8	Initialized	Initialized	Retained
	ECNTBF8	Initialized	Initialized	Retained
	GRBF8	Initialized	Initialized	Retained
	ECNTCF8	Initialized	Initialized	Retained
	GRAF8	Initialized	Initialized	Retained
	CDRF8	Initialized	Initialized	Retained
	GRCF8	Initialized	Initialized	Retained
	TCRF9	Initialized	Initialized	Retained
	TIERF9	Initialized	Initialized	Retained
	TSRF9	Initialized	Initialized	Retained
	ECNTAF9	Initialized	Initialized	Retained
	ECNTBF9	Initialized	Initialized	Retained
	GRBF9	Initialized	Initialized	Retained
	ECNTCF9	Initialized	Initialized	Retained
	GRAF9	Initialized	Initialized	Retained
	CDRF9	Initialized	Initialized	Retained
	GRCF9	Initialized	Initialized	Retained
	TCRF10	Initialized	Initialized	Retained
	TIERF10	Initialized	Initialized	Retained
	TSRF10	Initialized	Initialized	Retained
	ECNTAF10	Initialized	Initialized	Retained
	ECNTBF10	Initialized	Initialized	Retained
	GRBF10	Initialized	Initialized	Retained
	ECNTCF10	Initialized	Initialized	Retained
	GRAF10	Initialized	Initialized	Retained
	CDRF10	Initialized	Initialized	Retained
	GRCF10	Initialized	Initialized	Retained
	TCRF11	Initialized	Initialized	Retained
	TIERF11	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	Register Name	Power-On	Hardware Standby	Sleep
ATU-III	TSRF11	Initialized	Initialized	Retained
	ECNTAF11	Initialized	Initialized	Retained
	ECNTBF11	Initialized	Initialized	Retained
	GRBF11	Initialized	Initialized	Retained
	ECNTCF11	Initialized	Initialized	Retained
	GRAF11	Initialized	Initialized	Retained
	CDRF11	Initialized	Initialized	Retained
	GRCF11	Initialized	Initialized	Retained
	TCRF12	Initialized	Initialized	Retained
	TIERF12	Initialized	Initialized	Retained
	TSRF12	Initialized	Initialized	Retained
	ECNTAF12	Initialized	Initialized	Retained
	ECNTBF12	Initialized	Initialized	Retained
	GRBF12	Initialized	Initialized	Retained
	ECNTCF12	Initialized	Initialized	Retained
	GRAF12	Initialized	Initialized	Retained
	CDRF12	Initialized	Initialized	Retained
	GRCF12	Initialized	Initialized	Retained
	GRDF12	Initialized	Initialized	Retained
	TCRF13	Initialized	Initialized	Retained
	TIERF13	Initialized	Initialized	Retained
	TSRF13	Initialized	Initialized	Retained
	ECNTAF13	Initialized	Initialized	Retained
	ECNTBF13	Initialized	Initialized	Retained
	GRBF13	Initialized	Initialized	Retained
	ECNTCF13	Initialized	Initialized	Retained
	GRAF13	Initialized	Initialized	Retained
	CDRF13	Initialized	Initialized	Retained
	GRCF13	Initialized	Initialized	Retained
	GRDF13	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	**Register Name**	**Power-On**	**Hardware Standby**	**Sleep**
ATU-III	TCRF14	Initialized	Initialized	Retained
	TIERF14	Initialized	Initialized	Retained
	TSRF14	Initialized	Initialized	Retained
	ECNTAF14	Initialized	Initialized	Retained
	ECNTBF14	Initialized	Initialized	Retained
	GRBF14	Initialized	Initialized	Retained
	ECNTCF14	Initialized	Initialized	Retained
	GRAF14	Initialized	Initialized	Retained
	CDRF14	Initialized	Initialized	Retained
	GRCF14	Initialized	Initialized	Retained
	GRDF14	Initialized	Initialized	Retained
	TCRF15	Initialized	Initialized	Retained
	TIERF15	Initialized	Initialized	Retained
	TSRF15	Initialized	Initialized	Retained
	ECNTAF15	Initialized	Initialized	Retained
	ECNTBF15	Initialized	Initialized	Retained
	GRBF15	Initialized	Initialized	Retained
	ECNTCF15	Initialized	Initialized	Retained
	GRAF15	Initialized	Initialized	Retained
	CDRF15	Initialized	Initialized	Retained
	GRCF15	Initialized	Initialized	Retained
	GRDF15	Initialized	Initialized	Retained
	TCRF16	Initialized	Initialized	Retained
	TIERF16	Initialized	Initialized	Retained
	TSRF16	Initialized	Initialized	Retained
	ECNTAF16	Initialized	Initialized	Retained
	ECNTBF16	Initialized	Initialized	Retained
	GRBF16	Initialized	Initialized	Retained
	ECNTCF16	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	Register Name	Power-On	Hardware Standby	Sleep
ATU-III	GRAF16	Initialized	Initialized	Retained
	CDRF16	Initialized	Initialized	Retained
	GRCF16	Initialized	Initialized	Retained
	TCRF17	Initialized	Initialized	Retained
	TIERF17	Initialized	Initialized	Retained
	TSRF17	Initialized	Initialized	Retained
	ECNTAF17	Initialized	Initialized	Retained
	ECNTBF17	Initialized	Initialized	Retained
	GRBF17	Initialized	Initialized	Retained
	ECNTCF17	Initialized	Initialized	Retained
	GRAF17	Initialized	Initialized	Retained
	CDRF17	Initialized	Initialized	Retained
	GRCF17	Initialized	Initialized	Retained
	TCRF18	Initialized	Initialized	Retained
	TIERF18	Initialized	Initialized	Retained
	TSRF18	Initialized	Initialized	Retained
	ECNTAF18	Initialized	Initialized	Retained
	ECNTBF18	Initialized	Initialized	Retained
	GRBF18	Initialized	Initialized	Retained
	ECNTCF18	Initialized	Initialized	Retained
	GRAF18	Initialized	Initialized	Retained
	CDRF18	Initialized	Initialized	Retained
	GRCF18	Initialized	Initialized	Retained
	TCRF19	Initialized	Initialized	Retained
	TIERF19	Initialized	Initialized	Retained
	TSRF19	Initialized	Initialized	Retained
	ECNTAF19	Initialized	Initialized	Retained
	ECNTBF19	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	Register Name	Power-On	Hardware Standby	Sleep
ATU-III	GRBF19	Initialized	Initialized	Retained
	ECNTCF19	Initialized	Initialized	Retained
	GRAF19	Initialized	Initialized	Retained
	CDRF19	Initialized	Initialized	Retained
	GRCF19	Initialized	Initialized	Retained
	TCRF20	Initialized	Initialized	Retained
	TIERF20	Initialized	Initialized	Retained
	TSRF20	Initialized	Initialized	Retained
	ECNTAF20	Initialized	Initialized	Retained
	ECNTBF20	Initialized	Initialized	Retained
	GRBF20	Initialized	Initialized	Retained
	ECNTCF20	Initialized	Initialized	Retained
	GRAF20	Initialized	Initialized	Retained
	CDRF20	Initialized	Initialized	Retained
	GRCF20	Initialized	Initialized	Retained
	TCRF21	Initialized	Initialized	Retained
	TIERF21	Initialized	Initialized	Retained
	TSRF21	Initialized	Initialized	Retained
	ECNTAF21	Initialized	Initialized	Retained
	ECNTBF21	Initialized	Initialized	Retained
	GRBF21	Initialized	Initialized	Retained
	ECNTCF21	Initialized	Initialized	Retained
	GRAF21	Initialized	Initialized	Retained
	CDRF21	Initialized	Initialized	Retained
	GRCF21	Initialized	Initialized	Retained
	TCRF22	Initialized	Initialized	Retained
	TIERF22	Initialized	Initialized	Retained
	TSRF22	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	Register Name	Power-On	Hardware Standby	Sleep
ATU-III	ECNTAF22	Initialized	Initialized	Retained
	ECNTBF22	Initialized	Initialized	Retained
	GRBF22	Initialized	Initialized	Retained
	ECNTCF22	Initialized	Initialized	Retained
	GRAF22	Initialized	Initialized	Retained
	CDRF22	Initialized	Initialized	Retained
	GRCF22	Initialized	Initialized	Retained
	TCRF23	Initialized	Initialized	Retained
	TIERF23	Initialized	Initialized	Retained
	TSRF23	Initialized	Initialized	Retained
	ECNTAF23	Initialized	Initialized	Retained
	ECNTBF23	Initialized	Initialized	Retained
	GRBF23	Initialized	Initialized	Retained
	ECNTCF23	Initialized	Initialized	Retained
	GRAF23	Initialized	Initialized	Retained
	CDRF23	Initialized	Initialized	Retained
	GRCF23	Initialized	Initialized	Retained
	TCRF24	Initialized	Initialized	Retained
	TIERF24	Initialized	Initialized	Retained
	TSRF24	Initialized	Initialized	Retained
	ECNTAF24	Initialized	Initialized	Retained
	ECNTBF24	Initialized	Initialized	Retained
	GRBF24	Initialized	Initialized	Retained
	ECNTCF24	Initialized	Initialized	Retained
	GRAF24	Initialized	Initialized	Retained
	CDRF24	Initialized	Initialized	Retained
	GRCF24	Initialized	Initialized	Retained
	TCRF25	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	Register Name	Power-On	Hardware Standby	Sleep
ATU-III	TIERF25	Initialized	Initialized	Retained
	TSRF25	Initialized	Initialized	Retained
	ECNTAF25	Initialized	Initialized	Retained
	ECNTBF25	Initialized	Initialized	Retained
	GRBF25	Initialized	Initialized	Retained
	ECNTCF25	Initialized	Initialized	Retained
	GRAF25	Initialized	Initialized	Retained
	CDRF25	Initialized	Initialized	Retained
	GRCF25	Initialized	Initialized	Retained
	TCRF26	Initialized	Initialized	Retained
	TIERF26	Initialized	Initialized	Retained
	TSRF26	Initialized	Initialized	Retained
	ECNTAF26	Initialized	Initialized	Retained
	ECNTBF26	Initialized	Initialized	Retained
	GRBF26	Initialized	Initialized	Retained
	ECNTCF26	Initialized	Initialized	Retained
	GRAF26	Initialized	Initialized	Retained
	CDRF26	Initialized	Initialized	Retained
	GRCF26	Initialized	Initialized	Retained
	TCRF27	Initialized	Initialized	Retained
	TIERF27	Initialized	Initialized	Retained
	TSRF27	Initialized	Initialized	Retained
	ECNTAF27	Initialized	Initialized	Retained
	ECNTBF27	Initialized	Initialized	Retained
	GRBF27	Initialized	Initialized	Retained
	ECNTCF27	Initialized	Initialized	Retained
	GRAF27	Initialized	Initialized	Retained
	CDRF27	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	Register Name	Power-On	Hardware Standby	Sleep
ATU-III	GRCF27	Initialized	Initialized	Retained
	TSTRG	Initialized	Initialized	Retained
	TCRG0	Initialized	Initialized	Retained
	TSRG0	Initialized	Initialized	Retained
	TCNTG0	Initialized	Initialized	Retained
	OCRG0	Initialized	Initialized	Retained
	TCRG1	Initialized	Initialized	Retained
	TSRG1	Initialized	Initialized	Retained
	TCNTG1	Initialized	Initialized	Retained
	OCRG1	Initialized	Initialized	Retained
	TCRG2	Initialized	Initialized	Retained
	TSRG2	Initialized	Initialized	Retained
	TCNTG2	Initialized	Initialized	Retained
	OCRG2	Initialized	Initialized	Retained
	TCRG3	Initialized	Initialized	Retained
	TSRG3	Initialized	Initialized	Retained
	TCNTG3	Initialized	Initialized	Retained
	OCRG3	Initialized	Initialized	Retained
	TCRG4	Initialized	Initialized	Retained
	TSRG4	Initialized	Initialized	Retained
	TCNTG4	Initialized	Initialized	Retained
	OCRG4	Initialized	Initialized	Retained
	TCRG5	Initialized	Initialized	Retained
	TSRG5	Initialized	Initialized	Retained
	TCNTG5	Initialized	Initialized	Retained
	OCRG5	Initialized	Initialized	Retained
	TCRH	Initialized	Initialized	Retained
	TSRH	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	Register Name	Power-On	Hardware Standby	Sleep
ATU-III	TCNT1H	Initialized	Initialized	Retained
	OCR1H	Initialized	Initialized	Retained
	TCNT2H	Initialized	Initialized	Retained
	TSTRJ	Initialized	Initialized	Retained
	TCRJ0	Initialized	Initialized	Retained
	FCRJ0	Initialized	Initialized	Retained
	TSRJ0	Initialized	Initialized	Retained
	TIERJ0	Initialized	Initialized	Retained
	FDNRJ0	Initialized	Initialized	Retained
	NCNTJ0	Initialized	Initialized	Retained
	NCRJ0	Initialized	Initialized	Retained
	TCNTJ0	Initialized	Initialized	Retained
	OCRJ0	Initialized	Initialized	Retained
	FIFOJ0	Initialized	Initialized	Retained
	TCRJ1	Initialized	Initialized	Retained
	FCRJ1	Initialized	Initialized	Retained
	TSRJ1	Initialized	Initialized	Retained
	TIERJ1	Initialized	Initialized	Retained
	FDNRJ1	Initialized	Initialized	Retained
	NCNTJ1	Initialized	Initialized	Retained
	NCRJ1	Initialized	Initialized	Retained
	TCNTJ1	Initialized	Initialized	Retained
	OCRJ1	Initialized	Initialized	Retained
	FIFOJ1	Initialized	Initialized	Retained
WDT	WTCR	Initialized	Initialized	Retained
	WTCNT	Initialized	Initialized	Retained
	WTSR	Initialized	Initialized	Retained
	WRCR	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	Register Name	Power-On	Hardware Standby	Sleep
CMT	CMSTR	Initialized	Initialized	Retained
	CMCR_0	Initialized	Initialized	Retained
	CMSR_0	Initialized	Initialized	Retained
	CMCNT_0	Initialized	Initialized	Retained
	CMCOR_0	Initialized	Initialized	Retained
	CMCR_1	Initialized	Initialized	Retained
	CMSR_1	Initialized	Initialized	Retained
	CMCNT_1	Initialized	Initialized	Retained
	CMCOR_1	Initialized	Initialized	Retained
SCI	SCSMR1A	Initialized	Initialized	Retained
	SCBRR1A	Initialized	Initialized	Retained
	SCSCR1A	Initialized	Initialized	Retained
	SCTDR1A	Initialized	Initialized	Retained
	SCSSR1A	Initialized	Initialized	Retained
	SCRDR1A	Initialized	Initialized	Retained
	SCSMR1B	Initialized	Initialized	Retained
	SCBRR1B	Initialized	Initialized	Retained
	SCSCR1B	Initialized	Initialized	Retained
	SCTDR1B	Initialized	Initialized	Retained
	SCSSR1B	Initialized	Initialized	Retained
	SCRDR1B	Initialized	Initialized	Retained
	SCSMR1C	Initialized	Initialized	Retained
	SCBRR1C	Initialized	Initialized	Retained
	SCSCR1C	Initialized	Initialized	Retained
	SCTDR1C	Initialized	Initialized	Retained
	SCSSR1C	Initialized	Initialized	Retained
	SCRDR1C	Initialized	Initialized	Retained
	SCSMR1D	Initialized	Initialized	Retained
	SCBRR1D	Initialized	Initialized	Retained
	SCSCR1D	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	Register Name	Power-On	Hardware Standby	Sleep
SCI	SCTDR1D	Initialized	Initialized	Retained
	SCSSR1D	Initialized	Initialized	Retained
	SCRDR1D	Initialized	Initialized	Retained
	SCSMR1E	Initialized	Initialized	Retained
	SCBRR1E	Initialized	Initialized	Retained
	SCSCR1E	Initialized	Initialized	Retained
	SCTDR1E	Initialized	Initialized	Retained
	SCSSR1E	Initialized	Initialized	Retained
	SCRDR1E	Initialized	Initialized	Retained
RSPI	SPCRA	Initialized	Initialized	Retained
	SSLPA	Initialized	Initialized	Retained
	SPPCRA	Initialized	Initialized	Retained
	SPSRA	Initialized	Initialized	Retained
	SPDRA	Initialized	Initialized	Retained
	SPSCRA	Initialized	Initialized	Retained
	SPSSRA	Initialized	Initialized	Retained
	SPBRA	Initialized	Initialized	Retained
	SPCKDA	Initialized	Initialized	Retained
	SSLNDA	Initialized	Initialized	Retained
	SPNDA	Initialized	Initialized	Retained
	SPCMDA0	Initialized	Initialized	Retained
	SPCMDA1	Initialized	Initialized	Retained
	SPCMDA2	Initialized	Initialized	Retained
	SPCMDA3	Initialized	Initialized	Retained
	SPCMDA4	Initialized	Initialized	Retained
	SPCMDA5	Initialized	Initialized	Retained
	SPCMDA6	Initialized	Initialized	Retained
	SPCMDA7	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	Register Name	Power-On	Hardware Standby	Sleep
RSPI	SPCRB	Initialized	Initialized	Retained
	SSLPB	Initialized	Initialized	Retained
	SPPCRB	Initialized	Initialized	Retained
	SPSRB	Initialized	Initialized	Retained
	SPDRB	Initialized	Initialized	Retained
	SPSCRB	Initialized	Initialized	Retained
	SPSSRB	Initialized	Initialized	Retained
	SPBRB	Initialized	Initialized	Retained
	SPCKDB	Initialized	Initialized	Retained
	SSLNDB	Initialized	Initialized	Retained
	SPNDB	Initialized	Initialized	Retained
	SPCMDB0	Initialized	Initialized	Retained
	SPCMDB1	Initialized	Initialized	Retained
	SPCMDB2	Initialized	Initialized	Retained
	SPCMDB3	Initialized	Initialized	Retained
	SPCMDB4	Initialized	Initialized	Retained
	SPCMDB5	Initialized	Initialized	Retained
	SPCMDB6	Initialized	Initialized	Retained
	SPCMDB7	Initialized	Initialized	Retained
	SPCRC	Initialized	Initialized	Retained
	SSLPC	Initialized	Initialized	Retained
	SPPCRC	Initialized	Initialized	Retained
	SPSRC	Initialized	Initialized	Retained
	SPDRC	Initialized	Initialized	Retained
	SPSCRC	Initialized	Initialized	Retained
	SPSSRC	Initialized	Initialized	Retained
	SPBRC	Initialized	Initialized	Retained
	SPCKDC	Initialized	Initialized	Retained
	SSLNDC	Initialized	Initialized	Retained
	SPNDC	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	Register Name	Power-On	Hardware Standby	Sleep
RSPI	SPCMDC0	Initialized	Initialized	Retained
	SPCMDC1	Initialized	Initialized	Retained
	SPCMDC2	Initialized	Initialized	Retained
	SPCMDC3	Initialized	Initialized	Retained
	SPCMDC4	Initialized	Initialized	Retained
	SPCMDC5	Initialized	Initialized	Retained
	SPCMDC6	Initialized	Initialized	Retained
	SPCMDC7	Initialized	Initialized	Retained
RCAN-TL1 (RCAN_A)	MCR	Initialized	Initialized	Retained
	GSR	Initialized	Initialized	Retained
	BCR1	Initialized	Initialized	Retained
	BCR0	Initialized	Initialized	Retained
	IRR	Initialized	Initialized	Retained
	IMR	Initialized	Initialized	Retained
	TEC/REC	Initialized	Initialized	Retained
	TXPR1	Initialized	Initialized	Retained
	TXPR0	Initialized	Initialized	Retained
	TXCR1	Initialized	Initialized	Retained
	TXCR0	Initialized	Initialized	Retained
	TXACK1	Initialized	Initialized	Retained
	TXACK0	Initialized	Initialized	Retained
	ABACK1	Initialized	Initialized	Retained
	ABACK0	Initialized	Initialized	Retained
	RXPR1	Initialized	Initialized	Retained
	RXPR0	Initialized	Initialized	Retained
	RFPR1	Initialized	Initialized	Retained
	RFPR0	Initialized	Initialized	Retained
	MBIMR1	Initialized	Initialized	Retained
	MBIMR0	Initialized	Initialized	Retained

			Reset State	Power-Down State	
Module	**Register Name**		**Power-On**	**Hardware Standby**	**Sleep**
RCAN-TL1 (RCAN_A)	UMSR1		Initialized	Initialized	Retained
	UMSR0		Initialized	Initialized	Retained
	TTCR0		Initialized	Initialized	Retained
	CMAX_TEW		Initialized	Initialized	Retained
	RFTROFF		Initialized	Initialized	Retained
	TSR		Initialized	Initialized	Retained
	CCR		Initialized	Initialized	Retained
	TCNTR		Initialized	Initialized	Retained
	CYCTR		Initialized	Initialized	Retained
	RFMK		Initialized	Initialized	Retained
	TCMR0		Initialized	Initialized	Retained
	TCMR1		Initialized	Initialized	Retained
	TCMR2		Initialized	Initialized	Retained
	TTTSEL		Initialized	Initialized	Retained
	MB[0].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MB[1].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained

			Reset State	Power-Down State	
Module	Register Name		Power-On	Hardware Standby	Sleep
RCAN-TL1 (RCAN_A)	MB[1].	DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MB[2].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MB[3].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MB[4].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained

			Reset State	Power-Down State	
Module	**Register Name**		**Power-On**	**Hardware Standby**	**Sleep**
RCAN-TL1 (RCAN_A)	MB[4].	LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MB[5].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MB[6].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MB[7].	CONTROL0_H	Undefined	Undefined	Retained

			Reset State	Power-Down State	
Module	Register Name		Power-On	Hardware Standby	Sleep
RCAN-TL1 (RCAN_A)	MB[7].	CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MB[8].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MB[9].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained

			Reset State	Power-Down State	
Module	Register Name		Power-On	Hardware Standby	Sleep
RCAN-TL1 (RCAN_A)	MB[10].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MB[11].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MB[12].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained

Module	Register Name		Reset State Power-On	Power-Down State Hardware Standby	Sleep
RCAN-TL1 (RCAN_A)	MB[13].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MB[14].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MB[15].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained

			Reset State	Power-Down State	
Module	Register Name		Power-On	Hardware Standby	Sleep
RCAN-TL1 (RCAN_A)	MB[15].	CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MB[16].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
	MB[17].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
	MB[18].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained

			Reset State	Power-Down State	
Module	Register Name		Power-On	Hardware Standby	Sleep
RCAN-TL1 (RCAN_A)	MB[19].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
	MB[20].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
	MB[21].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
	MB[22].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained

			Reset State	Power-Down State	
Module	Register Name		Power-On	Hardware Standby	Sleep
RCAN-TL1 (RCAN_A)	MB[22].	LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
	MB[23].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
	MB[24].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TTT	Initialized	Initialized	Retained
		TTCONTROL	Initialized	Initialized	Retained

			Reset State	Power-Down State	
Module	**Register Name**		**Power-On**	**Hardware Standby**	**Sleep**
RCAN-TL1 (RCAN_A)	MB[25].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TTT	Initialized	Initialized	Retained
		TTCONTROL	Initialized	Initialized	Retained
	MB[26].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TTT	Initialized	Initialized	Retained
		TTCONTROL	Initialized	Initialized	Retained
	MB[27].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained

			Reset State	Power-Down State	
Module	Register Name		Power-On	Hardware Standby	Sleep
RCAN-TL1 (RCAN_A)	MB[27].	DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TTT	Initialized	Initialized	Retained
		TTCONTROL	Initialized	Initialized	Retained
	MB[28].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TTT	Initialized	Initialized	Retained
		TTCONTROL	Initialized	Initialized	Retained
	MB[29].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TTT	Initialized	Initialized	Retained
		TTCONTROL	Initialized	Initialized	Retained
	MB[30].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained

			Reset State	Power-Down State	
Module	Register Name		Power-On	Hardware Standby	Sleep
RCAN-TL1 (RCAN_A)	MB[30].	DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
		TTT	Initialized	Initialized	Retained
	MB[31].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MBESR		Initialized	Initialized	Retained
	MBECR		Initialized	Initialized	Retained
RCAN-TL1 (RCAN_B)	MCR		Initialized	Initialized	Retained
	GSR		Initialized	Initialized	Retained
	BCR1		Initialized	Initialized	Retained
	BCR0		Initialized	Initialized	Retained
	IRR		Initialized	Initialized	Retained
	IMR		Initialized	Initialized	Retained
	TEC/REC		Initialized	Initialized	Retained
	TXPR1		Initialized	Initialized	Retained
	TXPR0		Initialized	Initialized	Retained
	TXCR1		Initialized	Initialized	Retained

			Reset State	Power-Down State	
Module	Register Name		Power-On	Hardware Standby	Sleep
RCAN-TL1 (RCAN_B)	TXCR0		Initialized	Initialized	Retained
	TXACK1		Initialized	Initialized	Retained
	TXACK0		Initialized	Initialized	Retained
	ABACK1		Initialized	Initialized	Retained
	ABACK0		Initialized	Initialized	Retained
	RXPR1		Initialized	Initialized	Retained
	RXPR0		Initialized	Initialized	Retained
	RFPR1		Initialized	Initialized	Retained
	RFPR0		Initialized	Initialized	Retained
	MBIMR1		Initialized	Initialized	Retained
	MBIMR0		Initialized	Initialized	Retained
	UMSR1		Initialized	Initialized	Retained
	UMSR0		Initialized	Initialized	Retained
	TTCR0		Initialized	Initialized	Retained
	CMAX_TEW		Initialized	Initialized	Retained
	RFTROFF		Initialized	Initialized	Retained
	TSR		Initialized	Initialized	Retained
	CCR		Initialized	Initialized	Retained
	TCNTR		Initialized	Initialized	Retained
	CYCTR		Initialized	Initialized	Retained
	RFMK		Initialized	Initialized	Retained
	TCMR0		Initialized	Initialized	Retained
	TCMR1		Initialized	Initialized	Retained
	TCMR2		Initialized	Initialized	Retained
	TTTSEL		Initialized	Initialized	Retained
	MB[0].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained

			Reset State	Power-Down State	
Module	**Register Name**		**Power-On**	**Hardware Standby**	**Sleep**
RCAN-TL1 (RCAN_B)	MB[0].	DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MB[1].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MB[2].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MB[3].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained

			Reset State	Power-Down State	
Module	Register Name		Power-On	Hardware Standby	Sleep
RCAN-TL1 (RCAN_B)	MB[3].	DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MB[4].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MB[5].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MB[6].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained

			Reset State	Power-Down State	
Module	Register Name		Power-On	Hardware Standby	Sleep
RCAN-TL1 (RCAN_B)	MB[6].	DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MB[7].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MB[8].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MB[9].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained

			Reset State	Power-Down State	
Module	Register Name		Power-On	Hardware Standby	Sleep
RCAN-TL1 (RCAN_B)	MB[9].	DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MB[10].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MB[11].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MB[12].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained

			Reset State	Power-Down State	
Module	Register Name		Power-On	Hardware Standby	Sleep
RCAN-TL1 (RCAN_B)	MB[12].	LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MB[13].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MB[14].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MB[15].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained

			Reset State	Power-Down State	
Module	Register Name		Power-On	Hardware Standby	Sleep
RCAN-TL1 (RCAN_B)	MB[15].	LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MB[16].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
	MB[17].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
	MB[18].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained

			Reset State	Power-Down State	
Module	Register Name		Power-On	Hardware Standby	Sleep
RCAN-TL1 (RCAN_B)	MB[18].	LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
	MB[19].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
	MB[20].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
	MB[21].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained

			Reset State	Power-Down State	
Module	Register Name		Power-On	Hardware Standby	Sleep
RCAN-TL1 (RCAN_B)	MB[21].	DATA_01	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
	MB[22].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
	MB[23].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
	MB[24].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained

			Reset State	Power-Down State	
Module	Register Name		Power-On	Hardware Standby	Sleep
RCAN-TL1 (RCAN_B)	MB[24].	TTT	Initialized	Initialized	Retained
		TTCONTROL	Initialized	Initialized	Retained
	MB[25].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TTT	Initialized	Initialized	Retained
		TTCONTROL	Initialized	Initialized	Retained
	MB[26].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TTT	Initialized	Initialized	Retained
		TTCONTROL	Initialized	Initialized	Retained
	MB[27].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained

			Reset State	Power-Down State	
Module	Register Name		Power-On	Hardware Standby	Sleep
RCAN-TL1 (RCAN_B)	MB[27].	DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TTT	Initialized	Initialized	Retained
		TTCONTROL	Initialized	Initialized	Retained
	MB[28].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TTT	Initialized	Initialized	Retained
		TTCONTROL	Initialized	Initialized	Retained
	MB[29].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TTT	Initialized	Initialized	Retained
		TTCONTROL	Initialized	Initialized	Retained

			Reset State	Power-Down State	
Module	Register Name		Power-On	Hardware Standby	Sleep
RCAN-TL1 (RCAN_B)	MB[30].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
		TTT	Initialized	Initialized	Retained
	MB[31].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MBESR		Initialized	Initialized	Retained
	MBECR		Initialized	Initialized	Retained
RCAN-TL1 (RCAN_C)	MCR		Initialized	Initialized	Retained
	GSR		Initialized	Initialized	Retained
	BCR1		Initialized	Initialized	Retained
	BCR0		Initialized	Initialized	Retained
	IRR		Initialized	Initialized	Retained
	IMR		Initialized	Initialized	Retained
	TEC/REC		Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	Register Name	Power-On	Hardware Standby	Sleep
RCAN-TL1 (RCAN_C)	TXPR1	Initialized	Initialized	Retained
	TXPR0	Initialized	Initialized	Retained
	TXCR1	Initialized	Initialized	Retained
	TXCR0	Initialized	Initialized	Retained
	TXACK1	Initialized	Initialized	Retained
	TXACK0	Initialized	Initialized	Retained
	ABACK1	Initialized	Initialized	Retained
	ABACK0	Initialized	Initialized	Retained
	RXPR1	Initialized	Initialized	Retained
	RXPR0	Initialized	Initialized	Retained
	RFPR1	Initialized	Initialized	Retained
	RFPR0	Initialized	Initialized	Retained
	MBIMR1	Initialized	Initialized	Retained
	MBIMR0	Initialized	Initialized	Retained
	UMSR1	Initialized	Initialized	Retained
	UMSR0	Initialized	Initialized	Retained
	TTCR0	Initialized	Initialized	Retained
	CMAX_TEW	Initialized	Initialized	Retained
	RFTROFF	Initialized	Initialized	Retained
	TSR	Initialized	Initialized	Retained
	CCR	Initialized	Initialized	Retained
	TCNTR	Initialized	Initialized	Retained
	CYCTR	Initialized	Initialized	Retained
	RFMK	Initialized	Initialized	Retained
	TCMR0	Initialized	Initialized	Retained
	TCMR1	Initialized	Initialized	Retained
	TCMR2	Initialized	Initialized	Retained
	TTTSEL	Initialized	Initialized	Retained

			Reset State	Power-Down State	
Module	Register Name		Power-On	Hardware Standby	Sleep
RCAN-TL1 (RCAN_C)	MB[0].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MB[1].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MB[2].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained

			Reset State	Power-Down State	
Module	Register Name		Power-On	Hardware Standby	Sleep
RCAN-TL1 (RCAN_C)	MB[3].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MB[4].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MB[5].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained

			Reset State	Power-Down State	
Module	Register Name		Power-On	Hardware Standby	Sleep
RCAN-TL1 (RCAN_C)	MB[6].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MB[7].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MB[8].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained

			Reset State	Power-Down State	
Module	Register Name		Power-On	Hardware Standby	Sleep
RCAN-TL1 (RCAN_C)	MB[9].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MB[10].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MB[11].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained

			Reset State	Power-Down State	
Module	Register Name		Power-On	Hardware Standby	Sleep
RCAN-TL1 (RCAN_C)	MB[12].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MB[13].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MB[14].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained

			Reset State	**Power-Down State**	
Module	**Register Name**		**Power-On**	**Hardware Standby**	**Sleep**
RCAN-TL1 (RCAN_C)	MB[15].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MB[16].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
	MB[17].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
	MB[18].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained

			Reset State	Power-Down State	
Module	Register Name		Power-On	Hardware Standby	Sleep
RCAN-TL1 (RCAN_C)	MB[18].	LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
	MB[19].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
	MB[20].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
	MB[21].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained

			Reset State	Power-Down State	
Module	Register Name		Power-On	Hardware Standby	Sleep
RCAN-TL1 (RCAN_C)	MB[21].	DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
	MB[22].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
	MB[23].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
	MB[24].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained

			Reset State	Power-Down State	
Module	Register Name		Power-On	Hardware Standby	Sleep
RCAN-TL1 (RCAN_C)	MB[24].	TTT	Initialized	Initialized	Retained
		TTCONTROL	Initialized	Initialized	Retained
	MB[25].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TTT	Initialized	Initialized	Retained
		TTCONTROL	Initialized	Initialized	Retained
	MB[26].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TTT	Initialized	Initialized	Retained
		TTCONTROL	Initialized	Initialized	Retained
	MB[27].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained

			Reset State	Power-Down State	
Module	Register Name		Power-On	Hardware Standby	Sleep
RCAN-TL1 (RCAN_C)	MB[27].	DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TTT	Initialized	Initialized	Retained
		TTCONTROL	Initialized	Initialized	Retained
	MB[28].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TTT	Initialized	Initialized	Retained
		TTCONTROL	Initialized	Initialized	Retained
	MB[29].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TTT	Initialized	Initialized	Retained
		TTCONTROL	Initialized	Initialized	Retained

			Reset State	Power-Down State	
Module	Register Name		Power-On	Hardware Standby	Sleep
RCAN-TL1 (RCAN_C)	MB[30].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
		TTT	Initialized	Initialized	Retained
	MB[31].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MBESR		Initialized	Initialized	Retained
	MBECR		Initialized	Initialized	Retained
RCAN-TL1 (RCAN_D)	MCR		Initialized	Initialized	Retained
	GSR		Initialized	Initialized	Retained
	BCR1		Initialized	Initialized	Retained
	BCR0		Initialized	Initialized	Retained
	IRR		Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	**Register Name**	**Power-On**	**Hardware Standby**	**Sleep**
RCAN-TL1 (RCAN_D)	IMR	Initialized	Initialized	Retained
	TEC/REC	Initialized	Initialized	Retained
	TXPR1	Initialized	Initialized	Retained
	TXPR0	Initialized	Initialized	Retained
	TXCR1	Initialized	Initialized	Retained
	TXCR0	Initialized	Initialized	Retained
	TXACK1	Initialized	Initialized	Retained
	TXACK0	Initialized	Initialized	Retained
	ABACK1	Initialized	Initialized	Retained
	ABACK0	Initialized	Initialized	Retained
	RXPR1	Initialized	Initialized	Retained
	RXPR0	Initialized	Initialized	Retained
	RFPR1	Initialized	Initialized	Retained
	RFPR0	Initialized	Initialized	Retained
	MBIMR1	Initialized	Initialized	Retained
	MBIMR0	Initialized	Initialized	Retained
	UMSR1	Initialized	Initialized	Retained
	UMSR0	Initialized	Initialized	Retained
	TTCR0	Initialized	Initialized	Retained
	CMAX_TEW	Initialized	Initialized	Retained
	RFTROFF	Initialized	Initialized	Retained
	TSR	Initialized	Initialized	Retained
	CCR	Initialized	Initialized	Retained
	TCNTR	Initialized	Initialized	Retained
	CYCTR	Initialized	Initialized	Retained
	RFMK	Initialized	Initialized	Retained
	TCMR0	Initialized	Initialized	Retained
	TCMR1	Initialized	Initialized	Retained
	TCMR2	Initialized	Initialized	Retained
	TTTSEL	Initialized	Initialized	Retained

			Reset State	Power-Down State	
Module	Register Name		Power-On	Hardware Standby	Sleep
RCAN-TL1 (RCAN_D)	MB[0].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MB[1].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MB[2].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained

			Reset State	Power-Down State	
Module	Register Name		Power-On	Hardware Standby	Sleep
RCAN-TL1 (RCAN_D)	MB[3].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MB[4].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MB[5].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained

			Reset State	Power-Down State	
Module	Register Name		Power-On	Hardware Standby	Sleep
RCAN-TL1 (RCAN_D)	MB[6].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MB[7].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MB[8].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained

			Reset State	Power-Down State	
Module	Register Name		Power-On	Hardware Standby	Sleep
RCAN-TL1 (RCAN_D)	MB[9].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MB[10].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MB[11].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained

			Reset State	Power-Down State	
Module	Register Name		Power-On	Hardware Standby	Sleep
RCAN-TL1 (RCAN_D)	MB[12].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MB[13].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MB[14].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained

			Reset State	Power-Down State	
Module	Register Name		Power-On	Hardware Standby	Sleep
RCAN-TL1 (RCAN_D)	MB[15].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MB[16].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
	MB[17].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained

			Reset State	Power-Down State	
Module	**Register Name**		**Power-On**	**Hardware Standby**	**Sleep**
RCAN-TL1 (RCAN_D)	MB[18].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
	MB[19].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
	MB[20].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained

			Reset State	Power-Down State	
Module	Register Name		Power-On	Hardware Standby	Sleep
RCAN-TL1 (RCAN_D)	MB[21].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
	MB[22].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
	MB[23].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained

Module	Register Name		Reset State Power-On	Power-Down State Hardware Standby	Sleep
RCAN-TL1 (RCAN_D)	MB[24].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TTT	Initialized	Initialized	Retained
		TTCONTROL	Initialized	Initialized	Retained
	MB[25].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TTT	Initialized	Initialized	Retained
		TTCONTROL	Initialized	Initialized	Retained
	MB[26].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained

			Reset State	Power-Down State	
Module	Register Name		Power-On	Hardware Standby	Sleep
RCAN-TL1 (RCAN_D)	MB[26].	DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TTT	Initialized	Initialized	Retained
		TTCONTROL	Initialized	Initialized	Retained
	MB[27].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TTT	Initialized	Initialized	Retained
		TTCONTROL	Initialized	Initialized	Retained
	MB[28].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TTT	Initialized	Initialized	Retained
		TTCONTROL	Initialized	Initialized	Retained
	MB[29].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained

			Reset State	Power-Down State	
Module	Register Name		Power-On	Hardware Standby	Sleep
RCAN-TL1 (RCAN_D)	MB[29].	LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TTT	Initialized	Initialized	Retained
		TTCONTROL	Initialized	Initialized	Retained
	MB[30].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained
		CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
		TTT	Initialized	Initialized	Retained
	MB[31].	CONTROL0_H	Undefined	Undefined	Retained
		CONTROL0_L	Undefined	Undefined	Retained
		LAFM0	Undefined	Undefined	Retained
		LAFM1	Undefined	Undefined	Retained
		DATA_01	Undefined	Undefined	Retained
		DATA_23	Undefined	Undefined	Retained
		DATA_45	Undefined	Undefined	Retained
		DATA_67	Undefined	Undefined	Retained

			Reset State	Power-Down State	
Module	Register Name		Power-On	Hardware Standby	Sleep
RCAN-TL1 (RCAN_D)	MB[31].	CONTROL1	Initialized	Initialized	Retained
		TIMESTAMP	Initialized	Initialized	Retained
	MBESR		Initialized	Initialized	Retained
	MBECR		Initialized	Initialized	Retained
FlexRay	FXROC		Initialized	Initialized	Retained
	FXROS		Initialized	Initialized	Retained
	FXRTISR		Initialized	Initialized	Retained
	FXRTIER		Initialized	Initialized	Retained
	FRLCK		Initialized	Initialized	Retained
	FREIR		Initialized	Initialized	Retained
	FRSIR		Initialized	Initialized	Retained
	FREILS		Initialized	Initialized	Retained
	FRSILS		Initialized	Initialized	Retained
	FREIES		Initialized	Initialized	Retained
	FREIER		Initialized	Initialized	Retained
	FRSIES		Initialized	Initialized	Retained
	FRSIER		Initialized	Initialized	Retained
	FRILE		Initialized	Initialized	Retained
	FRT0C		Initialized	Initialized	Retained
	FRT1C		Initialized	Initialized	Retained
	FRSTPW1		Initialized	Initialized	Retained
	FRSTPW2		Initialized	Initialized	Retained
	FRSUCC1		Initialized	Initialized	Retained
	FRSUCC2		Initialized	Initialized	Retained
	FRSUCC3		Initialized	Initialized	Retained
	FRNEMC		Initialized	Initialized	Retained
	FRPRTC1		Initialized	Initialized	Retained
	FRPRTC2		Initialized	Initialized	Retained
	FRMHDC		Initialized	Initialized	Retained
	FRGTUC1		Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	Register Name	Power-On	Hardware Standby	Sleep
FlexRay	FRGTUC2	Initialized	Initialized	Retained
	FRGTUC3	Initialized	Initialized	Retained
	FRGTUC4	Initialized	Initialized	Retained
	FRGTUC5	Initialized	Initialized	Retained
	FRGTUC6	Initialized	Initialized	Retained
	FRGTUC7	Initialized	Initialized	Retained
	FRGTUC8	Initialized	Initialized	Retained
	FRGTUC9	Initialized	Initialized	Retained
	FRGTUC10	Initialized	Initialized	Retained
	FRGTUC11	Initialized	Initialized	Retained
	FRCCSV	Initialized	Initialized	Retained
	FRCCEV	Initialized	Initialized	Retained
	FRSCV	Initialized	Initialized	Retained
	FRMTCCV	Initialized	Initialized	Retained
	FRRCV	Initialized	Initialized	Retained
	FROCV	Initialized	Initialized	Retained
	FRSFS	Initialized	Initialized	Retained
	FRSWNIT	Initialized	Initialized	Retained
	FRACS	Initialized	Initialized	Retained
	FRESID1	Initialized	Initialized	Retained
	FRESID2	Initialized	Initialized	Retained
	FRESID3	Initialized	Initialized	Retained
	FRESID4	Initialized	Initialized	Retained
	FRESID5	Initialized	Initialized	Retained
	FRESID6	Initialized	Initialized	Retained
	FRESID7	Initialized	Initialized	Retained
	FRESID8	Initialized	Initialized	Retained
	FRESID9	Initialized	Initialized	Retained
	FRESID10	Initialized	Initialized	Retained
	FRESID11	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	Register Name	Power-On	Hardware Standby	Sleep
FlexRay	FRESID12	Initialized	Initialized	Retained
	FRESID13	Initialized	Initialized	Retained
	FRESID14	Initialized	Initialized	Retained
	FRESID15	Initialized	Initialized	Retained
	FROSID1	Initialized	Initialized	Retained
	FROSID2	Initialized	Initialized	Retained
	FROSID3	Initialized	Initialized	Retained
	FROSID4	Initialized	Initialized	Retained
	FROSID5	Initialized	Initialized	Retained
	FROSID6	Initialized	Initialized	Retained
	FROSID7	Initialized	Initialized	Retained
	FROSID8	Initialized	Initialized	Retained
	FROSID9	Initialized	Initialized	Retained
	FROSID10	Initialized	Initialized	Retained
	FROSID11	Initialized	Initialized	Retained
	FROSID12	Initialized	Initialized	Retained
	FROSID13	Initialized	Initialized	Retained
	FROSID14	Initialized	Initialized	Retained
	FROSID15	Initialized	Initialized	Retained
	FRNMV1	Initialized	Initialized	Retained
	FRNMV2	Initialized	Initialized	Retained
	FRNMV3	Initialized	Initialized	Retained
	FRMRC	Initialized	Initialized	Retained
	FRFRF	Initialized	Initialized	Retained
	FRFRFM	Initialized	Initialized	Retained
	FRFCL	Initialized	Initialized	Retained
	FRMHDS	Initialized	Initialized	Retained
	FRLDTS	Initialized	Initialized	Retained
	FRFSR	Initialized	Initialized	Retained
	FRMHDF	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	Register Name	Power-On	Hardware Standby	Sleep
FlexRay	FRTXRQ1	Initialized	Initialized	Retained
	FRTXRQ2	Initialized	Initialized	Retained
	FRTXRQ3	Initialized	Initialized	Retained
	FRTXRQ4	Initialized	Initialized	Retained
	FRNDAT1	Initialized	Initialized	Retained
	FRNDAT2	Initialized	Initialized	Retained
	FRNDAT3	Initialized	Initialized	Retained
	FRNDAT4	Initialized	Initialized	Retained
	FRMBSC1	Initialized	Initialized	Retained
	FRMBSC2	Initialized	Initialized	Retained
	FRMBSC3	Initialized	Initialized	Retained
	FRMBSC4	Initialized	Initialized	Retained
	FRWRDS1	Initialized	Initialized	Retained
	FRWRDS2	Initialized	Initialized	Retained
	FRWRDS3	Initialized	Initialized	Retained
	FRWRDS4	Initialized	Initialized	Retained
	FRWRDS5	Initialized	Initialized	Retained
	FRWRDS6	Initialized	Initialized	Retained
	FRWRDS7	Initialized	Initialized	Retained
	FRWRDS8	Initialized	Initialized	Retained
	FRWRDS9	Initialized	Initialized	Retained
	FRWRDS10	Initialized	Initialized	Retained
	FRWRDS11	Initialized	Initialized	Retained
	FRWRDS12	Initialized	Initialized	Retained
	FRWRDS13	Initialized	Initialized	Retained
	FRWRDS14	Initialized	Initialized	Retained
	FRWRDS15	Initialized	Initialized	Retained
	FRWRDS16	Initialized	Initialized	Retained
	FRWRDS17	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	Register Name	Power-On	Hardware Standby	Sleep
FlexRay	FRWRDS18	Initialized	Initialized	Retained
	FRWRDS19	Initialized	Initialized	Retained
	FRWRDS20	Initialized	Initialized	Retained
	FRWRDS21	Initialized	Initialized	Retained
	FRWRDS22	Initialized	Initialized	Retained
	FRWRDS23	Initialized	Initialized	Retained
	FRWRDS24	Initialized	Initialized	Retained
	FRWRDS25	Initialized	Initialized	Retained
	FRWRDS26	Initialized	Initialized	Retained
	FRWRDS27	Initialized	Initialized	Retained
	FRWRDS28	Initialized	Initialized	Retained
	FRWRDS29	Initialized	Initialized	Retained
	FRWRDS30	Initialized	Initialized	Retained
	FRWRDS31	Initialized	Initialized	Retained
	FRWRDS32	Initialized	Initialized	Retained
	FRWRDS33	Initialized	Initialized	Retained
	FRWRDS34	Initialized	Initialized	Retained
	FRWRDS35	Initialized	Initialized	Retained
	FRWRDS36	Initialized	Initialized	Retained
	FRWRDS37	Initialized	Initialized	Retained
	FRWRDS38	Initialized	Initialized	Retained
	FRWRDS39	Initialized	Initialized	Retained
	FRWRDS40	Initialized	Initialized	Retained
	FRWRDS41	Initialized	Initialized	Retained
	FRWRDS42	Initialized	Initialized	Retained
	FRWRDS43	Initialized	Initialized	Retained
	FRWRDS44	Initialized	Initialized	Retained
	FRWRDS45	Initialized	Initialized	Retained
	FRWRDS46	Initialized	Initialized	Retained
	FRWRDS47	Initialized	Initialized	Retained
	FRWRDS48	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	**Register Name**	**Power-On**	**Hardware Standby**	**Sleep**
FlexRay	FRWRDS49	Initialized	Initialized	Retained
	FRWRDS50	Initialized	Initialized	Retained
	FRWRDS51	Initialized	Initialized	Retained
	FRWRDS52	Initialized	Initialized	Retained
	FRWRDS53	Initialized	Initialized	Retained
	FRWRDS54	Initialized	Initialized	Retained
	FRWRDS55	Initialized	Initialized	Retained
	FRWRDS56	Initialized	Initialized	Retained
	FRWRDS57	Initialized	Initialized	Retained
	FRWRDS58	Initialized	Initialized	Retained
	FRWRDS59	Initialized	Initialized	Retained
	FRWRDS60	Initialized	Initialized	Retained
	FRWRDS61	Initialized	Initialized	Retained
	FRWRDS62	Initialized	Initialized	Retained
	FRWRDS63	Initialized	Initialized	Retained
	FRWRDS64	Initialized	Initialized	Retained
	FRWRHS1	Initialized	Initialized	Retained
	FRWRHS2	Initialized	Initialized	Retained
	FRWRHS3	Initialized	Initialized	Retained
	FRIBCM	Initialized	Initialized	Retained
	FRIBCR	Initialized	Initialized	Retained
	FRRDDS1	Initialized	Initialized	Retained
	FRRDDS2	Initialized	Initialized	Retained
	FRRDDS3	Initialized	Initialized	Retained
	FRRDDS4	Initialized	Initialized	Retained
	FRRDDS5	Initialized	Initialized	Retained
	FRRDDS6	Initialized	Initialized	Retained
	FRRDDS7	Initialized	Initialized	Retained
	FRRDDS8	Initialized	Initialized	Retained
	FRRDDS9	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	Register Name	Power-On	Hardware Standby	Sleep
FlexRay	FRRDDS10	Initialized	Initialized	Retained
	FRRDDS11	Initialized	Initialized	Retained
	FRRDDS12	Initialized	Initialized	Retained
	FRRDDS13	Initialized	Initialized	Retained
	FRRDDS14	Initialized	Initialized	Retained
	FRRDDS15	Initialized	Initialized	Retained
	FRRDDS16	Initialized	Initialized	Retained
	FRRDDS17	Initialized	Initialized	Retained
	FRRDDS18	Initialized	Initialized	Retained
	FRRDDS19	Initialized	Initialized	Retained
	FRRDDS20	Initialized	Initialized	Retained
	FRRDDS21	Initialized	Initialized	Retained
	FRRDDS22	Initialized	Initialized	Retained
	FRRDDS23	Initialized	Initialized	Retained
	FRRDDS24	Initialized	Initialized	Retained
	FRRDDS25	Initialized	Initialized	Retained
	FRRDDS26	Initialized	Initialized	Retained
	FRRDDS27	Initialized	Initialized	Retained
	FRRDDS28	Initialized	Initialized	Retained
	FRRDDS29	Initialized	Initialized	Retained
	FRRDDS30	Initialized	Initialized	Retained
	FRRDDS31	Initialized	Initialized	Retained
	FRRDDS32	Initialized	Initialized	Retained
	FRRDDS33	Initialized	Initialized	Retained
	FRRDDS34	Initialized	Initialized	Retained
	FRRDDS35	Initialized	Initialized	Retained
	FRRDDS36	Initialized	Initialized	Retained
	FRRDDS37	Initialized	Initialized	Retained
	FRRDDS38	Initialized	Initialized	Retained
	FRRDDS39	Initialized	Initialized	Retained
	FRRDDS40	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	Register Name	Power-On	Hardware Standby	Sleep
FlexRay	FRRDDS41	Initialized	Initialized	Retained
	FRRDDS42	Initialized	Initialized	Retained
	FRRDDS43	Initialized	Initialized	Retained
	FRRDDS44	Initialized	Initialized	Retained
	FRRDDS45	Initialized	Initialized	Retained
	FRRDDS46	Initialized	Initialized	Retained
	FRRDDS47	Initialized	Initialized	Retained
	FRRDDS48	Initialized	Initialized	Retained
	FRRDDS49	Initialized	Initialized	Retained
	FRRDDS50	Initialized	Initialized	Retained
	FRRDDS51	Initialized	Initialized	Retained
	FRRDDS52	Initialized	Initialized	Retained
	FRRDDS53	Initialized	Initialized	Retained
	FRRDDS54	Initialized	Initialized	Retained
	FRRDDS55	Initialized	Initialized	Retained
	FRRDDS56	Initialized	Initialized	Retained
	FRRDDS57	Initialized	Initialized	Retained
	FRRDDS58	Initialized	Initialized	Retained
	FRRDDS59	Initialized	Initialized	Retained
	FRRDDS60	Initialized	Initialized	Retained
	FRRDDS61	Initialized	Initialized	Retained
	FRRDDS62	Initialized	Initialized	Retained
	FRRDDS63	Initialized	Initialized	Retained
	FRRDDS64	Initialized	Initialized	Retained
	FRRDHS1	Initialized	Initialized	Retained
	FRRDHS2	Initialized	Initialized	Retained
	FRRDHS3	Initialized	Initialized	Retained
	FRMBS	Initialized	Initialized	Retained
	FROBCM	Initialized	Initialized	Retained
	FROBCR	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	Register Name	Power-On	Hardware Standby	Sleep
ADC (ADC_A)	ADRD0	Initialized	Initialized	Retained
	ADR0	Initialized	Initialized	Retained
	ADR1	Initialized	Initialized	Retained
	ADR2	Initialized	Initialized	Retained
	ADR3	Initialized	Initialized	Retained
	ADR4	Initialized	Initialized	Retained
	ADR5	Initialized	Initialized	Retained
	ADR6	Initialized	Initialized	Retained
	ADR7	Initialized	Initialized	Retained
	ADR8	Initialized	Initialized	Retained
	ADR9	Initialized	Initialized	Retained
	ADR10	Initialized	Initialized	Retained
	ADR11	Initialized	Initialized	Retained
	ADR12	Initialized	Initialized	Retained
	ADR13	Initialized	Initialized	Retained
	ADR14	Initialized	Initialized	Retained
	ADR15	Initialized	Initialized	Retained
	ADR16	Initialized	Initialized	Retained
	ADR17	Initialized	Initialized	Retained
	ADR18	Initialized	Initialized	Retained
	ADR19	Initialized	Initialized	Retained
	ADR20	Initialized	Initialized	Retained
	ADR21	Initialized	Initialized	Retained
	ADR22	Initialized	Initialized	Retained
	ADR23	Initialized	Initialized	Retained
	ADR24	Initialized	Initialized	Retained
	ADR25	Initialized	Initialized	Retained
	ADR26	Initialized	Initialized	Retained
	ADR27	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	Register Name	Power-On	Hardware Standby	Sleep
ADC (ADC_B)	ADRD1	Initialized	Initialized	Retained
	ADR40	Initialized	Initialized	Retained
	ADR41	Initialized	Initialized	Retained
	ADR42	Initialized	Initialized	Retained
	ADR43	Initialized	Initialized	Retained
	ADR44	Initialized	Initialized	Retained
	ADR45	Initialized	Initialized	Retained
	ADR46	Initialized	Initialized	Retained
	ADR47	Initialized	Initialized	Retained
	ADR48	Initialized	Initialized	Retained
ADC (ADC_A)	ADCSR0	Initialized	Initialized	Retained
ADC (ADC_B)	ADCSR1	Initialized	Initialized	Retained
ADC (ADC_A)	ADREF0	Initialized	Initialized	Retained
ADC (ADC_B)	ADREF1	Initialized	Initialized	Retained
ADC (ADC_A)	ADTRE0	Initialized	Initialized	Retained
ADC (ADC_B)	ADTRE1	Initialized	Initialized	Retained
ADC (ADC_A)	ADTRF0	Initialized	Initialized	Retained
ADC (ADC_B)	ADTRF1	Initialized	Initialized	Retained
ADC (ADC_A)	ADTRS0	Initialized	Initialized	Retained
ADC (ADC_B)	ADTRS1	Initialized	Initialized	Retained
ADC (ADC_A)	ADSTRG0	Initialized	Initialized	Retained
ADC (ADC_B)	ADSTRG1	Initialized	Initialized	Retained
ADC (ADC_A)	ADTRD0	Initialized	Initialized	Retained
ADC (ADC_B)	ADTRD1	Initialized	Initialized	Retained
ADC (ADC_A)	ADADS0	Initialized	Initialized	Retained
	ADADS2	Initialized	Initialized	Retained
	ADADS3	Initialized	Initialized	Retained
	ADADS4	Initialized	Initialized	Retained
ADC (ADC_B)	ADADS1	Initialized	Initialized	Retained
	ADADS5	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	**Register Name**	**Power-On**	**Hardware Standby**	**Sleep**
ADC (ADC_A)	ADADC0	Initialized	Initialized	Retained
ADC (ADC_B)	ADADC1	Initialized	Initialized	Retained
ADC (ADC_A)	ADANS0	Initialized	Initialized	Retained
	ADANS1	Initialized	Initialized	Retained
ADC (ADC_B)	ADANS3	Initialized	Initialized	Retained
ADC (ADC_A)	ADCER0	Initialized	Initialized	Retained
ADC (ADC_B)	ADCER1	Initialized	Initialized	Retained
JTAG	SDIR	Undefined	Undefined	Retained
	SDID	Retained	Retained	Retained
	SDBPR	Undefined	Undefined	Retained
	SDBSR	Undefined	Undefined	Retained
AUD-II	AUCSR	Initialized	Initialized	Retained
	AUWASR	Undefined	Undefined	Retained
	AUWAER	Undefined	Undefined	Retained
	AUWBSR	Undefined	Undefined	Retained
	AUWBER	Undefined	Undefined	Retained
	AUECSR	Initialized	Initialized	Retained
PFC	PAIOR	Initialized	Initialized	Retained
	PACR4	Initialized	Initialized	Retained
	PACR3	Initialized	Initialized	Retained
	PACR2	Initialized	Initialized	Retained
	PACR1	Initialized	Initialized	Retained
	PBIOR	Initialized	Initialized	Retained
	PBCR4	Initialized	Initialized	Retained
	PBCR3	Initialized	Initialized	Retained
	PBCR2	Initialized	Initialized	Retained
	PBCR1	Initialized	Initialized	Retained
	PCIOR	Initialized	Initialized	Retained
	PCCR4	Initialized	Initialized	Retained
	PCCR3	Initialized	Initialized	Retained
	PCCR2	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	Register Name	Power-On	Hardware Standby	Sleep
PFC	PCCR1	Initialized	Initialized	Retained
	PDIOR	Initialized	Initialized	Retained
	PDCR2	Initialized	Initialized	Retained
	PDCR1	Initialized	Initialized	Retained
	PEIOR	Initialized	Initialized	Retained
	PECR2	Initialized	Initialized	Retained
	PECR1	Initialized	Initialized	Retained
	PFIOR	Initialized	Initialized	Retained
	PFCR2	Initialized	Initialized	Retained
	PFCR1	Initialized	Initialized	Retained
	PGIOR	Initialized	Initialized	Retained
	PGCR2	Initialized	Initialized	Retained
	PGCR1	Initialized	Initialized	Retained
	PHIOR	Initialized	Initialized	Retained
	PHCR	Initialized	Initialized	Retained
	PJIOR	Initialized	Initialized	Retained
	PJCR2	Initialized	Initialized	Retained
	PJCR1	Initialized	Initialized	Retained
	PKIOR	Initialized	Initialized	Retained
	PKCR2	Initialized	Initialized	Retained
	PKCR1	Initialized	Initialized	Retained
	PLIOR	Initialized	Initialized	Retained
	PLCR2	Initialized	Initialized	Retained
	PLCR1	Initialized	Initialized	Retained
	PACR4A	Initialized	Initialized	Retained
	PACR3A	Initialized	Initialized	Retained
	PACR2A	Initialized	Initialized	Retained
	PBCR4A	Initialized	Initialized	Retained
	PBCR3A	Initialized	Initialized	Retained
	PBCR2A	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	Register Name	Power-On	Hardware Standby	Sleep
PFC	PBCR1A	Initialized	Initialized	Retained
	PCCR4A	Initialized	Initialized	Retained
	PCCR3A	Initialized	Initialized	Retained
	PCCR2A	Initialized	Initialized	Retained
	PDCR2A	Initialized	Initialized	Retained
	PECR1A	Initialized	Initialized	Retained
	PFCR2A	Initialized	Initialized	Retained
	PJCR3A	Initialized	Initialized	Retained
	PJCR2A	Initialized	Initialized	Retained
	PJCR1A	Initialized	Initialized	Retained
I/O port	PADR	Initialized	Initialized	Retained
	PAPR	Pin value	Pin value	Retained
	PAIR	Initialized	Initialized	Retained
	PBDR	Initialized	Initialized	Retained
	PBPR	Pin value	Pin value	Retained
	PBIR	Initialized	Initialized	Retained
	PBDSR	Initialized	Initialized	Retained
	PBPSR	Initialized	Initialized	Retained
	PCDR	Initialized	Initialized	Retained
	PCPR	Pin value	Pin value	Retained
	PCIR	Initialized	Initialized	Retained
	PCDSR	Initialized	Initialized	Retained
	PCPSR	Initialized	Initialized	Retained
	PDDR	Initialized	Initialized	Retained
	PDPR	Pin value	Pin value	Retained
	PDIR	Initialized	Initialized	Retained
	PEDR	Initialized	Initialized	Retained
	PEPR	Pin value	Pin value	Retained
	PEIR	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	Register Name	Power-On	Hardware Standby	Sleep
I/O port	PEDSR	Initialized	Initialized	Retained
	PFDR	Initialized	Initialized	Retained
	PFPR	Pin value	Pin value	Retained
	PFIR	Initialized	Initialized	Retained
	PFDSR	Initialized	Initialized	Retained
	PFPSR	Initialized	Initialized	Retained
	PGDR	Initialized	Initialized	Retained
	PGPR	Pin value	Pin value	Retained
	PGIR	Initialized	Initialized	Retained
	PGDSR	Initialized	Initialized	Retained
	PGER	Initialized	Initialized	Retained
	PHDR	Initialized	Initialized	Retained
	PHPR	Pin value	Pin value	Retained
	PJDR	Initialized	Initialized	Retained
	PJPR	Pin value	Pin value	Retained
	PJIR	Initialized	Initialized	Retained
	PJDSR	Initialized	Initialized	Retained
	PJPSR	Initialized	Initialized	Retained
	PKDR	Initialized	Initialized	Retained
	PKPR	Pin value	Pin value	Retained
	PKIR	Initialized	Initialized	Retained
	PKDSR	Initialized	Initialized	Retained
	PKPSR	Initialized	Initialized	Retained
	PLDR	Initialized	Initialized	Retained
	PLPR	Pin value	Pin value	Retained
	PLIR	Initialized	Initialized	Retained
	CKCR	Initialized	Initialized	Retained
MISG	MISRCDR	Initialized	Initialized	Retained
	MISR	Initialized	Initialized	Retained
	MISR2	Initialized	Initialized	Retained
	MISRCR	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	Register Name	Power-On	Hardware Standby	Sleep
ROM/IFS/ EEPROM	FPMON	Initialized	Initialized	Retained
	FMODR	Initialized	Initialized	Retained
	FASTAT	Initialized	Initialized	Retained
	FAEINT	Initialized	Initialized	Retained
	ROMMAT	Initialized	Initialized	Retained
	EEPRE0	Initialized	Initialized	Retained
	EEPRE1	Initialized	Initialized	Retained
	EEPWE0	Initialized	Initialized	Retained
	EEPWE1	Initialized	Initialized	Retained
	FCURAME	Initialized	Initialized	Retained
	IFSCMR	Initialized	Initialized	Retained
	FSTATR0	Initialized	Initialized	Retained
	FSTATR1	Initialized	Initialized	Retained
	FENTRYR	Initialized	Initialized	Retained
	FPROTR	Initialized	Initialized	Retained
	FRESETR	Initialized	Initialized	Retained
	FCUSTATR	Initialized	Initialized	Retained
	FCMDR	Initialized	Initialized	Retained
	FRAMECCR	Initialized	Initialized	Retained
	IFSSR	Initialized	Initialized	Retained
	FCPSR	Initialized	Initialized	Retained
	EEPBCCNT	Initialized	Initialized	Retained
	FPESTAT	Initialized	Initialized	Retained
	EEPBCSTAT	Initialized	Initialized	Retained
	EEPMAT	Initialized	Initialized	Retained
ROMC	RCCR	Initialized	Initialized	Retained
	RCCR2	Initialized	Initialized	Retained
RAM	RAMEN0	Initialized	Initialized	Retained
	RAMWEN0	Initialized	Initialized	Retained

		Reset State	Power-Down State	
Module	**Register Name**	**Power-On**	**Hardware Standby**	**Sleep**
RAM	RAMECC	Initialized	Initialized	Retained
	RAMERR	Initialized	Initialized	Retained
	RAMINT	Initialized	Initialized	Retained
	RAMACYC	Initialized	Initialized	Retained
	RAMEN1	Initialized	Initialized	Retained
	RAMWEN1	Initialized	Initialized	Retained
Power-down mode	STBCR	Initialized	Initialized	Retained

Section 34 Electrical Characteristics

34.1 Absolute Maximum Ratings

Table 34.1 shows the absolute maximum ratings.

Table 34.1 Absolute Maximum Ratings

Item			Symbol	Rating	Unit	Remarks
Power supply voltage*1	V_{CC} and PLLV_{CC} pins		V_{CC}	–0.3 to +4.3	V	
	PV_{CC}1 and PV_{CC}2 pins		PV_{CC}	–0.3 to +6.5	V	
Input voltage	V_{CC} power supply related pins	EXTAL, JTAG, AUD-II pins	Vin	–0.3 to V_{CC} + 0.3	V	Refer to table 34.2, Correspondence between Power Supply Names and Pins
		$\overline{\text{HSTBY}}$, $\overline{\text{RES}}$, NMI, FWE, MD0, MD1, MD2 pins*4	Vin	–0.3 to 5.5 + 0.3	V	
		ASEMD, MD4, MD3, MD_CLK1, MD_CLK0, MD_CLKP pins*5	Vin	–0.3 to V_{CC} + 0.3	V	
		ASEMD, MD4, MD3, MD_CLK1, MD_CLK0, MD_CLKP pins*3	Vin	–0.3 to 5.5 + 0.3	V	
	PV_{CC} power supply related pins		Vin	–0.3 to PV_{CC} + 0.3	V	
Analog supply voltage			AV_{CC}	–0.3 to +6.5	V	
Analog reference voltage			AVREFH	–0.3 to AV_{CC} + 0.3	V	AVREFH > AVREFL
			AVREFL	–0.3 to AV_{SS} + 0.3	V	
Analog input voltage			V_{AN}	–0.3 to AV_{CC} + 0.3	V	

Item		Symbol	Rating	Unit	Remarks
V_{ss} differential voltage		V_{ss} – PLLV_{ss}	–0.1 to 0.1	V	
		V_{ss} – AV_{ss}	–0.1 to 0.1	V	
		PLLV_{ss} – AV_{ss}	–0.1 to 0.1	V	
Maximum input current per pin	Digital input pins	Imax	–25 to +25	mA	One pin at a time
	Analog input pins	Imax	–25 to +25	mA	
Operating temperature*[2]		Topr	–40 to +125	°C	
Storage temperature		Tstg	–55 to +125	°C	Before assembly

[Operating precautions]

Operating the LSI in excess of the absolute maximum ratings may result in permanent damage. The two power supply voltages of PV_{CC} of 5V and V_{CC} of 3.3V may be used simultaneously with the LSI. Be sure to use the LSI in compliance with the connection of power pins, combination conditions of applicable power supply voltages, voltage applicable to each pin, and conditions of output voltage, as specified in the manual. Connecting a non-specified power supply or using the LSI at an incorrect voltage may result in permanent damage of the LSI or the system that contains the LSI.

Notes: 1. Do not apply any power supply voltage to the V_{CL} pin. Connect to GND through an external capacitor.

2. When this LSI is used in the range of 85°C to 125°C, the accumulated operating time must be within 3000 hours.

3. When higher level voltage than V_{CC} + 0.3 V is input, it is recommended that the voltage input is connected to the pins through the combined resistance of all the pins, 200 kΩ or higher.
However, although the resistance value can be lowered depending on the sink current of external V_{CC} (3-V system) regulators, the combined resistance value must be 33 kΩ at least.
When the resistance value is modified, make sure that the V_{CC} voltage does not exceed 3.6 V in the standby state.

4. When the ASEMD, MD4, MD3, MD_CLK1, MD_CLK0, and MD_CLKP pins are input in 5-V tolerant, it is recommended that the $\overline{\text{HSTBY}}$, $\overline{\text{RES}}$, NMI, FWE, MD0, MD1, and MD2 pins are input in 5-V amplitude.

5. When the ASEMD, MD4, MD3, MD_CLK1, MD_CLK0, and MD_CLKP pins are input in 3.3-V, the $\overline{\text{HSTBY}}$, $\overline{\text{RES}}$, NMI, FWE, MD0, MD1, and MD2 pins can be input in either 3.3-V or 5-V amplitude.

34.2 DC Characteristics

Table 34.2 shows the correspondence between power supply names and pins.

Table 34.2 Correspondence between Power Supply Names and Pins

Pin No.	Power Supply Name on Pin	Symbol	User Pin				Power Supply Name in Circuit	Permissible Input Voltage (V)	Input Buffer Type	Output Driving Ability	Notes
			Function 1	Function 2	Function 3	Function 4					
A18		ASEMD					V_{cc}	V_{cc} + 0.3, 5.5 + 0.3*	Schmitt A		
C14		MD4					V_{cc}	V_{cc} + 0.3, 5.5 + 0.3*	Schmitt A		
C13		MD3					V_{cc}	V_{cc} + 0.3, 5.5 + 0.3*	Schmitt A		
D12		MD2					V_{cc}	5.5 + 0.3	Schmitt A		
A12		MD1					V_{cc}	5.5 + 0.3	Schmitt A		
C12		MD0					V_{cc}	5.5 + 0.3	Schmitt A		
A11		FWE					V_{cc}	5.5 + 0.3	Schmitt A		
B16		MD_CLK1					V_{cc}	V_{cc} + 0.3, 5.5 + 0.3*	Schmitt A		
D14		MD_CLK0					V_{cc}	V_{cc} + 0.3, 5.5 + 0.3*	Schmitt A		
B15		MD_CLKP					V_{cc}	V_{cc} + 0.3, 5.5 + 0.3*	Schmitt A		
D11		$\overline{\text{HSTBY}}$					V_{cc}	5.5 + 0.3	Schmitt A		
B12		$\overline{\text{RES}}$					V_{cc}	5.5 + 0.3	Schmitt A		
C15		NMI					V_{cc}	5.5 + 0.3	Schmitt A		
A14		EXTAL					V_{cc}	V_{cc} + 0.3	CMOS		
A15		XTAL					V_{cc}				
A13		CK					V_{cc}				
B11		$\overline{\text{WDTOVF}}$					V_{cc}				
B17		$\overline{\text{TRST}}$					V_{cc}	V_{cc} + 0.3	Schmitt A		
B18		TCK					V_{cc}	V_{cc} + 0.3	TTL		
D15		TMS					V_{cc}	V_{cc} + 0.3	TTL / Schmitt B		
A19		TDI					V_{cc}	V_{cc} + 0.3	TTL / Schmitt B		

Pin No.	Power Supply Name on Pin	Symbol	User Pin Function 1	 Function 2	 Function 3	 Function 4	Power Supply Name in Circuit	Permissible Input Voltage (V)	Input Buffer Type	Output Driving Ability	Notes
C16		TDO					V_{cc}				
C17		AUDMD					V_{cc}	V_{cc} + 0.3	Schmitt A		
D16		$\overline{\text{AUDRST}}$					V_{cc}	V_{cc} + 0.3	Schmitt A		
G18		AUDCK					V_{cc}	V_{cc} + 0.3	TTL		
D18		$\overline{\text{AUDSYNC}}$					V_{cc}	V_{cc} + 0.3	TTL		
D17		AUDATA3					V_{cc}	V_{cc} + 0.3	TTL		
F18		AUDATA2					V_{cc}	V_{cc} + 0.3	TTL		
E17		AUDATA1					V_{cc}	V_{cc} + 0.3	TTL		
E18		AUDATA0					V_{cc}	V_{cc} + 0.3	TTL		
C18			PA0	A0			$PV_{cc}1$	$PV_{cc}1$ + 0.3	Schmitt B		
B19			PA1	A1			$PV_{cc}1$	$PV_{cc}1$ + 0.3	Schmitt B		
C19			PA2	A2			$PV_{cc}1$	$PV_{cc}1$ + 0.3	Schmitt B		
E19			PA3	A3			$PV_{cc}1$	$PV_{cc}1$ + 0.3	Schmitt B		
D19			PA4	A4			$PV_{cc}1$	$PV_{cc}1$ + 0.3	Schmitt B		
G17			PA5	A5			$PV_{cc}1$	$PV_{cc}1$ + 0.3	Schmitt B		
F19			PA6	A6			$PV_{cc}1$	$PV_{cc}1$ + 0.3	Schmitt B		
B20			PA7	A7			$PV_{cc}1$	$PV_{cc}1$ + 0.3	Schmitt B		
H18			PA8	A8			$PV_{cc}1$	$PV_{cc}1$ + 0.3	Schmitt B		
C20			PA9	A9			$PV_{cc}1$	$PV_{cc}1$ + 0.3	Schmitt B		
J18			PA10	A10			$PV_{cc}1$	$PV_{cc}1$ + 0.3	Schmitt B		
G19			PA11	A11			$PV_{cc}1$	$PV_{cc}1$ + 0.3	Schmitt B		
D20			PA12	A12			$PV_{cc}1$	$PV_{cc}1$ + 0.3	Schmitt B		
E20			PA13	A13			$PV_{cc}1$	$PV_{cc}1$ + 0.3	Schmitt B		
H19			PA14	A14			$PV_{cc}1$	$PV_{cc}1$ + 0.3	Schmitt B		
F20			PA15	A15			$PV_{cc}1$	$PV_{cc}1$ + 0.3	Schmitt B		
G20			PB0	A16	MOSIA		$PV_{cc}1$	$PV_{cc}1$ + 0.3	Schmitt B	O	
K18			PB1	A17	MISOA		$PV_{cc}1$	$PV_{cc}1$ + 0.3	Schmitt B	O	
J19			PB2	A18	MOSIB		$PV_{cc}1$	$PV_{cc}1$ + 0.3	Schmitt B	O	
L18			PB3	A19	MISOB		$PV_{cc}1$	$PV_{cc}1$ + 0.3	Schmitt B	O	

Pin No.	Power Supply Name on Pin	Symbol	User Pin				Power Supply Name in Circuit	Permissible Input Voltage (V)	Input Buffer Type	Output Driving Ability	Notes
			Function 1	Function 2	Function 3	Function 4					
H20			PB4	A20	CTx_B	TIF6	PV$_{cc}$1	PV$_{cc}$1 + 0.3	Schmitt B	O	
J20			PB5	A21	CRx_B	TIF7	PV$_{cc}$1	PV$_{cc}$1 + 0.3	Schmitt B		
K19			PB6	$\overline{\text{WE0}}$			PV$_{cc}$1	PV$_{cc}$1 + 0.3	Schmitt B	O	
L17			PB7	$\overline{\text{WE1}}$			PV$_{cc}$1	PV$_{cc}$1 + 0.3	Schmitt B	O	
K20			PB8	$\overline{\text{WAIT}}$	TOE20		PV$_{cc}$1	PV$_{cc}$1 + 0.3	Schmitt B		
M19			PB9	$\overline{\text{RD}}$			PV$_{cc}$1	PV$_{cc}$1 + 0.3	Schmitt B	O	
M18			PB10	$\overline{\text{CS0}}$			PV$_{cc}$1	PV$_{cc}$1 + 0.3	Schmitt B	O	
M20			PB11	$\overline{\text{CS1}}$	TOE21		PV$_{cc}$1	PV$_{cc}$1 + 0.3	Schmitt B	O	
N20			PB12	$\overline{\text{CS2}}$	RSPCKA		PV$_{cc}$1	PV$_{cc}$1 + 0.3	Schmitt B	O	
N19			PB13	$\overline{\text{CS3}}$	RSPCKB		PV$_{cc}$1	PV$_{cc}$1 + 0.3	Schmitt B	O	
P20			PB14	RD/$\overline{\text{WR}}$			PV$_{cc}$1	PV$_{cc}$1 + 0.3	Schmitt B		
N18			PC0	D0			PV$_{cc}$1	PV$_{cc}$1 + 0.3	TTL		
R20			PC1	D1			PV$_{cc}$1	PV$_{cc}$1 + 0.3	TTL		
P19			PC2	D2			PV$_{cc}$1	PV$_{cc}$1 + 0.3	TTL		
N17			PC3	D3			PV$_{cc}$1	PV$_{cc}$1 + 0.3	TTL		
T20			PC4	D4			PV$_{cc}$1	PV$_{cc}$1 + 0.3	TTL		
R19			PC5	D5			PV$_{cc}$1	PV$_{cc}$1 + 0.3	TTL		
T19			PC6	D6			PV$_{cc}$1	PV$_{cc}$1 + 0.3	TTL		
P18			PC7	D7			PV$_{cc}$1	PV$_{cc}$1 + 0.3	TTL		
U20			PC8	D8			PV$_{cc}$1	PV$_{cc}$1 + 0.3	TTL	O	
U19			PC9	D9			PV$_{cc}$1	PV$_{cc}$1 + 0.3	TTL		
R18			PC10	D10			PV$_{cc}$1	PV$_{cc}$1 + 0.3	TTL	O	
V20			PC11	D11			PV$_{cc}$1	PV$_{cc}$1 + 0.3	TTL	O	
W20			PC12	D12			PV$_{cc}$1	PV$_{cc}$1 + 0.3	TTL	O	
V19			PC13	D13			PV$_{cc}$1	PV$_{cc}$1 + 0.3	TTL	O	
U18			PC14	D14			PV$_{cc}$1	PV$_{cc}$1 + 0.3	TTL	O	
T18			PC15	D15			PV$_{cc}$1	PV$_{cc}$1 + 0.3	TTL	O	
B6			PD0	TIOC00	TIOC31		PV$_{cc}$2	PV$_{cc}$2 + 0.3	Schmitt B		
B7			PD1	TIOC01	TOE20		PV$_{cc}$2	PV$_{cc}$2 + 0.3	Schmitt B		

Pin No.	Power Supply Name on Pin	Symbol	User Pin				Power Supply Name in Circuit	Permissible Input Voltage (V)	Input Buffer Type	Output Driving Ability	Notes
			Function 1	Function 2	Function 3	Function 4					
A4			PD2	TIOC02	TOE21	TOE52	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
A5			PD3	TIOC03	TOE22	TOE53	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
C9			PD4	TIOC10	TIOC32	TOE52	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
B8			PD5	TIOC11	TOE23	TOE40	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
A6			PD6	TIOC12		TOE41	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
A7			PD7	TIOC13		TOE42	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
A8			PD8	TIOC20	TIOC33	TOE53	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
B9			PD9	TIOC21	TIF0B	TOE43	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
C10			PD10	TIOC22	TIF1B	TOE50	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
A9			PD11	TIOC23	TIF2B	TOE51	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
D10			PD12	TCLKA	TIOC41	TIJ0	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
C11			PD13	TCLKB		TIJ1	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
D4			PE0	TIA00			$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
D5			PE1	TIA01	TIOC42	TIOC40	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
C6			PE2	TIA02	TIOC43	TIOC30	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
C5			PE3	TIA03			$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
C4			PE4	TIA04			$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
B3			PE5	TIA05			$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
B4			PE6	TOE00	CTx_B		$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
D6			PE7	TOE01	CRx_B		$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
B5			PE8	TOE02			$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
C7			PE9	TOE03			$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
A2			PE10	TOE10			$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
A3			PE11	TOE11			$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
C8			PE12	TOE12			$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
D8			PE13	TOE13			$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
V11			PF0	TOD00B		TIF6	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
Y13			PF1	TOD01B		TIF7	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		

Pin No.	Power Supply Name on Pin	Symbol	User Pin				Power Supply Name in Circuit	Permissible Input Voltage (V)	Input Buffer Type	Output Driving Ability	Notes
			Function 1	Function 2	Function 3	Function 4					
Y14			PF2	TOD02B		TIF8	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
W13			PF3	TOD03B		TIF9	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
V12			PF4	TOD10B		TIF10	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
Y15			PF5	TOD11B		TIF11	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
Y16			PF6	TOD12B		TIF12	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
U13			PF7	TOD13B		TIF13	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
W14			PF8	TOD20B		TIF14	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
Y17			PF9	TOD21B		TIF15	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
V13			PF10	TOD22B		TIF16	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
Y18			PF11	TOD23B		TIF17	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
V14			PF12	TOD30B		TIF18	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
W15			PF13	TOD31B		TIF19	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
Y19			PF14	TOD32B	CTx_B	TxD_A	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
W16			PF15	TOD33B	CRx_B	RxD_A	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
G1			PG0	TOD00A	SSLA0		$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
H3			PG1	TOD01A	SSLA1		$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
H2			PG2	TOD02A	SSLA2		$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
H1			PG3	TOD03A	SSLA3		$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
J1			PG4	TOD10A	SSLA4	SSLB3	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
J3			PG5	TOD11A	SSLA5	SSLC3	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
J2			PG6	TOD12A	SSLB0		$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
K4			PG7	TOD13A	SSLB1		$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
L1			PG8	TOD20A	SSLB2	TIF6	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
K3			PG9	TOD21A	SSLC0	TIF7	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
L2			PG10	TOD22A	SSLC1	TIF8	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
L3			PG11	TOD23A	SSLC2	TIF9	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
M1			PG12	TOD30A	SSLA4	TIF10	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
M4			PG13	TOD31A	SSLA5	TIF11	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
M2			PG14	TOD32A	SSLA6	TIF12	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
N1			PG15	TOD33A	SSLA7	TIF13	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	

Pin No.	Power Supply Name on Pin	Symbol	User Pin				Power Supply Name in Circuit	Permissible Input Voltage (V)	Input Buffer Type	Output Driving Ability	Notes
			Function 1	Function 2	Function 3	Function 4					
B1			PH0		ADTRG_A	TIF0A	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
E3			PH1		ADTRG_B	TIF1A	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
F4			PH2			TIF2A	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
B2			PH3			TIF3	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
D3			PH4			TIF4	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
C3			PH5			TIF5	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
V18			PJ0	TxD_A	CTx_A	CTx_A & CTx_B	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
V17			PJ1	RxD_A	CRx_A	CRx_A & CRx_B	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
V16			PJ2	TxD_A	CTx_C	CTx_A & CTx_B & CTx_C	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
U17			PJ3	RxD_A	CRx_C	CRx_A & CRx_B & CRx_C	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
T17			PJ4	SCK_A	ADEND_B	TIJ0	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
V15			PJ5	TxD_A			$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
W19			PJ6	RxD_A			$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
U15			PJ7	SCK_B	ADEND_A	TIJ1	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
W18			PJ8	TxD_B			$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
W17			PJ9	RxD_B			$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
G2			PK0	SCK_C	RSPCKA	$\overline{\text{UBCTRG}}$	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
H4			PK1	TxD_C	MOSIA		$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
F1			PK2	RxD_C	MISOA		$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
E1			PK3	SCK_D	RSPCKB		$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
G3			PK4	TxD_D	MOSIB		$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
F2			PK5	RxD_D	MISOB		$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
D1			PK6	SCK_E			$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
C1			PK7	TxD_E			$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	

Pin No.	Power Supply Name on Pin	Symbol	User Pin				Power Supply Name in Circuit	Permissible Input Voltage (V)	Input Buffer Type	Output Driving Ability	Notes
			Function 1	Function 2	Function 3	Function 4					
E2			PK8	RxD_E			$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
F3			PK9		RSPCKC		$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
D2			PK10		MOSIC		$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
C2			PK11		MISOC		$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
P3			PL0		$\overline{\text{IRQ0}}$		$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
P4			PL1	TOE20	$\overline{\text{IRQ1}}$	$\overline{\text{POD}}$	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
N3			PL2	TOE21	$\overline{\text{IRQ2}}$		$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
R2			PL3	TOE22	$\overline{\text{IRQ3}}$		$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
P2			PL4	TOE23	$\overline{\text{IRQ4}}$		$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
R1			PL5	TOE30	$\overline{\text{IRQ5}}$		$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
N2			PL6	TOE31	$\overline{\text{IRQ6}}$		$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
M3			PL7	TOE32	$\overline{\text{IRQ7}}$		$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
P1			PL8	TOE33			$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
W11		AN_A0					AV_{cc}	$AV_{cc} + 0.3$	Analog		
U11		AN_A1					AV_{cc}	$AV_{cc} + 0.3$	Analog		
Y11		AN_A2					AV_{cc}	$AV_{cc} + 0.3$	Analog		
V10		AN_A3					AV_{cc}	$AV_{cc} + 0.3$	Analog		
U10		AN_A4					AV_{cc}	$AV_{cc} + 0.3$	Analog		
Y9		AN_A5					AV_{cc}	$AV_{cc} + 0.3$	Analog		
V9		AN_A6					AV_{cc}	$AV_{cc} + 0.3$	Analog		
W9		AN_A7					AV_{cc}	$AV_{cc} + 0.3$	Analog		
U9		AN_A8					AV_{cc}	$AV_{cc} + 0.3$	Analog		
Y8		AN_A9					AV_{cc}	$AV_{cc} + 0.3$	Analog		
Y7		AN_A10					AV_{cc}	$AV_{cc} + 0.3$	Analog		
Y6		AN_A11					AV_{cc}	$AV_{cc} + 0.3$	Analog		
W8		AN_A12					AV_{cc}	$AV_{cc} + 0.3$	Analog		
V8		AN_A13					AV_{cc}	$AV_{cc} + 0.3$	Analog		
Y5		AN_A14					AV_{cc}	$AV_{cc} + 0.3$	Analog		

Pin No.	Power Supply Name on Pin	Symbol	User Pin				Power Supply Name in Circuit	Permissible Input Voltage (V)	Input Buffer Type	Output Driving Ability	Notes
			Function 1	Function 2	Function 3	Function 4					
V7		AN_A15					AV$_{cc}$	AV$_{cc}$ + 0.3	Analog		
W7		AN_A16					AV$_{cc}$	AV$_{cc}$ + 0.3	Analog		
W6		AN_A17					AV$_{cc}$	AV$_{cc}$ + 0.3	Analog		
W5		AN_A18					AV$_{cc}$	AV$_{cc}$ + 0.3	Analog		
U7		AN_A19					AV$_{cc}$	AV$_{cc}$ + 0.3	Analog		
V6		AN_A20					AV$_{cc}$	AV$_{cc}$ + 0.3	Analog		
U6		AN_A21					AV$_{cc}$	AV$_{cc}$ + 0.3	Analog		
V5		AN_A22					AV$_{cc}$	AV$_{cc}$ + 0.3	Analog		
U5		AN_A23					AV$_{cc}$	AV$_{cc}$ + 0.3	Analog		
Y2		AN_A24					AV$_{cc}$	AV$_{cc}$ + 0.3	Analog		
W2		AN_A25					AV$_{cc}$	AV$_{cc}$ + 0.3	Analog		
V3		AN_A26					AV$_{cc}$	AV$_{cc}$ + 0.3	Analog		
V4		AN_A27					AV$_{cc}$	AV$_{cc}$ + 0.3	Analog		
Y3		AVREFH_A					AV$_{cc}$	AV$_{cc}$ + 0.3	Analog		
W3		AVREFL_A					AV$_{cc}$	AV$_{ss}$ + 0.3	Analog		
T3		AN_B40					AV$_{cc}$	AV$_{cc}$ + 0.3	Analog		
U4		AN_B41					AV$_{cc}$	AV$_{cc}$ + 0.3	Analog		
U3		AN_B42					AV$_{cc}$	AV$_{cc}$ + 0.3	Analog		
T4		AN_B43					AV$_{cc}$	AV$_{cc}$ + 0.3	Analog		
R3		AN_B44					AV$_{cc}$	AV$_{cc}$ + 0.3	Analog		
V2		AN_B45					AV$_{cc}$	AV$_{cc}$ + 0.3	Analog		
R4		AN_B46					AV$_{cc}$	AV$_{cc}$ + 0.3	Analog		
U2		AN_B47					AV$_{cc}$	AV$_{cc}$ + 0.3	Analog		
T2		AN_B48					AV$_{cc}$	AV$_{cc}$ + 0.3	Analog		
W1		AVREFH_B					AV$_{cc}$	AV$_{cc}$ + 0.3	Analog		
V1		AVREFL_B					AV$_{cc}$	AV$_{ss}$ + 0.3	Analog		
A17	PLLV$_{cc}$										

Pin No.	Power Supply Name on Pin	Symbol	User Pin				Power Supply Name in Circuit	Permissible Input Voltage (V)	Input Buffer Type	Output Driving Ability	Notes
			Function 1	Function 2	Function 3	Function 4					
A16	PLLV$_{SS}$										
B10	V$_{CL}$										
K2	V$_{CL}$										
L19	V$_{CL}$										
W12	V$_{CL}$										
B14	V$_{CC}$										
D7	V$_{CC}$										
D13	V$_{CC}$										
F17	V$_{CC}$										
G4	V$_{CC}$										
L4	V$_{CC}$										
M17	V$_{CC}$										
U8	V$_{CC}$										
U14	V$_{CC}$										
H17	PV$_{CC}$1										
J17	PV$_{CC}$1										
K17	PV$_{CC}$1										
P17	PV$_{CC}$1										
R17	PV$_{CC}$1										
D9	PV$_{CC}$2										
E4	PV$_{CC}$2										
J4	PV$_{CC}$2										
N4	PV$_{CC}$2										
U12	PV$_{CC}$2										
U16	PV$_{CC}$2										
A1	V$_{SS}$										
A10	V$_{SS}$										
A20	V$_{SS}$										

Pin No.	Power Supply Name on Pin	Symbol	User Pin				Power Supply Name in Circuit	Permissible Input Voltage (V)	Input Buffer Type	Output Driving Ability	Notes
			Function 1	Function 2	Function 3	Function 4					
B13	V_{ss}										
J9	V_{ss}										
J10	V_{ss}										
J11	V_{ss}										
J12	V_{ss}										
K1	V_{ss}										
K9	V_{ss}										
K10	V_{ss}										
K11	V_{ss}										
K12	V_{ss}										
L9	V_{ss}										
L10	V_{ss}										
L11	V_{ss}										
L12	V_{ss}										
L20	V_{ss}										
M9	V_{ss}										
M10	V_{ss}										
M11	V_{ss}										
M12	V_{ss}										
Y12	V_{ss}										
Y20	V_{ss}										
T1	AV_{cc}										
W10	AV_{cc}										
Y4	AV_{cc}										
U1	AV_{ss}										
W4	AV_{ss}										
Y10	AV_{ss}										
Y1	NC										

			User Pin								
Pin No.	Power Supply Name on Pin	Symbol	Function 1	Function 10	Function 11	Function 12	Power Supply Name in Circuit	Permissible Input Voltage (V)	Input Buffer Type	Output Driving Ability	Notes
C18			PA0				$PV_{cc}1$	$PV_{cc}1 + 0.3$	Schmitt B		
B19			PA1				$PV_{cc}1$	$PV_{cc}1 + 0.3$	Schmitt B		
C19			PA2				$PV_{cc}1$	$PV_{cc}1 + 0.3$	Schmitt B		
E19			PA3				$PV_{cc}1$	$PV_{cc}1 + 0.3$	Schmitt B		
D19			PA4	TOD00B			$PV_{cc}1$	$PV_{cc}1 + 0.3$	Schmitt B		
G17			PA5	TOD01B	TIF0A		$PV_{cc}1$	$PV_{cc}1 + 0.3$	Schmitt B		
F19			PA6	TOD02B	TIF1A		$PV_{cc}1$	$PV_{cc}1 + 0.3$	Schmitt B		
B20			PA7	TOD03B	TIF2A		$PV_{cc}1$	$PV_{cc}1 + 0.3$	Schmitt B		
H18			PA8	TOD10B			$PV_{cc}1$	$PV_{cc}1 + 0.3$	Schmitt B		
C20			PA9	TOD11B	TIF0B		$PV_{cc}1$	$PV_{cc}1 + 0.3$	Schmitt B		
J18			PA10	TOD12B	TIF1B		$PV_{cc}1$	$PV_{cc}1 + 0.3$	Schmitt B		
G19			PA11	TOD13B	TIF2B		$PV_{cc}1$	$PV_{cc}1 + 0.3$	Schmitt B		
D20			PA12	TIA00			$PV_{cc}1$	$PV_{cc}1 + 0.3$	Schmitt B		
E20			PA13	TIA01			$PV_{cc}1$	$PV_{cc}1 + 0.3$	Schmitt B		
H19			PA14	TIA02			$PV_{cc}1$	$PV_{cc}1 + 0.3$	Schmitt B		
F20			PA15	TIA03			PVCC1	$PV_{cc}1 + 0.3$	Schmitt B		
G20			PB0			FRTxD_B	$PV_{cc}1$	$PV_{cc}1 + 0.3$	Schmitt B	O	
K18			PB1			FRRxD_B	$PV_{cc}1$	$PV_{cc}1 + 0.3$	Schmitt B	O	
J19			PB2	TIA04		SSLA6	$PV_{cc}1$	$PV_{cc}1 + 0.3$	Schmitt B	O	
L18			PB3	TIA05		SSLA7	$PV_{cc}1$	$PV_{cc}1 + 0.3$	Schmitt B	O	
H20			PB4	TIF26		TxD_B	$PV_{cc}1$	$PV_{cc}1 + 0.3$	Schmitt B	O	
J20			PB5	TIF27		RxD_B	$PV_{cc}1$	$PV_{cc}1 + 0.3$	Schmitt B		
K19			PB6			SCK_B	$PV_{cc}1$	$PV_{cc}1 + 0.3$	Schmitt B	O	
L17			PB7			TxD_A	$PV_{cc}1$	$PV_{cc}1 + 0.3$	Schmitt B	O	
K20			PB8	TIF20		RxD_A	$PV_{cc}1$	$PV_{cc}1 + 0.3$	Schmitt B		
M19			PB9			SSLB0	$PV_{cc}1$	$PV_{cc}1 + 0.3$	Schmitt B	O	
M18			PB10			SSLB1	$PV_{cc}1$	$PV_{cc}1 + 0.3$	Schmitt B	O	
M20			PB11	TIF21		SSLB2	$PV_{cc}1$	$PV_{cc}1 + 0.3$	Schmitt B	O	

Pin No.	Power Supply Name on Pin	Symbol	User Pin Function 1	Function 10	Function 11	Function 12	Power Supply Name in Circuit	Permissible Input Voltage (V)	Input Buffer Type	Output Driving Ability	Notes
N20			PB12			FREN_B	$PV_{cc}1$	$PV_{cc}1 + 0.3$	Schmitt B	O	
N19			PB13	TIF22			$PV_{cc}1$	$PV_{cc}1 + 0.3$	Schmitt B	O	
P20			PB14				$PV_{cc}1$	$PV_{cc}1 + 0.3$	Schmitt B		
N18			PC0				$PV_{cc}1$	$PV_{cc}1 + 0.3$	TTL		
R20			PC1				$PV_{cc}1$	$PV_{cc}1 + 0.3$	TTL		
P19			PC2				$PV_{cc}1$	$PV_{cc}1 + 0.3$	TTL		
N17			PC3				$PV_{cc}1$	$PV_{cc}1 + 0.3$	TTL		
T20			PC4				$PV_{cc}1$	$PV_{cc}1 + 0.3$	TTL		
R19			PC5				$PV_{cc}1$	$PV_{cc}1 + 0.3$	TTL		
T19			PC6			ADTRG_A	$PV_{cc}1$	$PV_{cc}1 + 0.3$	TTL		
P18			PC7			ADEND_A	$PV_{cc}1$	$PV_{cc}1 + 0.3$	TTL		
U20			PC8	TOD00A	TxD_A	CTx_A	$PV_{cc}1$	$PV_{cc}1 + 0.3$	TTL	O	
U19			PC9	TOD01A	RxD_A	CRx_A	$PV_{cc}1$	$PV_{cc}1 + 0.3$	TTL		
R18			PC10	TOD02A	SCK_A		$PV_{cc}1$	$PV_{cc}1 + 0.3$	TTL	O	
V20			PC11	TOD03A		SSLB3	$PV_{cc}1$	$PV_{cc}1 + 0.3$	TTL	O	
W20			PC12	TOD10A		SSLA0	$PV_{cc}1$	$PV_{cc}1 + 0.3$	TTL	O	
V19			PC13	TOD11A		SSLA1	$PV_{cc}1$	$PV_{cc}1 + 0.3$	TTL	O	
U18			PC14	TOD12A		SSLA2	$PV_{cc}1$	$PV_{cc}1 + 0.3$	TTL	O	
T18			PC15	TOD13A		SSLA3	$PV_{cc}1$	$PV_{cc}1 + 0.3$	TTL	O	
B6			PD0				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
B7			PD1				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
A4			PD2				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
A5			PD3				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
C9			PD4				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
B8			PD5				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
A6			PD6				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
A7			PD7				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
A8			PD8				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		

Pin No.	Power Supply Name on Pin	Symbol	User Pin				Power Supply Name in Circuit	Permissible Input Voltage (V)	Input Buffer Type	Output Driving Ability	Notes
			Function 1	Function 10	Function 11	Function 12					
B9			PD9				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
C10			PD10				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
A9			PD11				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
D10			PD12				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
C11			PD13	TOE60			$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
D4			PE0				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
D5			PE1				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
C6			PE2				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
C5			PE3	TOE61	TIF23		$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
C4			PE4	TOE62	TIF24		$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
B3			PE5	TOE63	TIF25		$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
B4			PE6				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
D6			PE7				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
B5			PE8				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
C7			PE9				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
A2			PE10				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
A3			PE11				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
C8			PE12				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
D8			PE13				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
V11			PF0				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
Y13			PF1				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
Y14			PF2				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
W13			PF3				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
V12			PF4				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
Y15			PF5				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
Y16			PF6				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
U13			PF7				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
W14			PF8				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		

Pin No.	Power Supply Name on Pin	Symbol	User Pin Function 1	 Function 10	 Function 11	 Function 12	Power Supply Name in Circuit	Permissible Input Voltage (V)	Input Buffer Type	Output Driving Ability	Notes
Y17			PF9				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
V13			PF10				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
Y18			PF11				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
V14			PF12				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
W15			PF13				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
Y19			PF14			FRTxD_B	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
W16			PF15			FRRxD_B	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
G1			PG0				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
H3			PG1				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
H2			PG2				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
H1			PG3				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
J1			PG4				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
J3			PG5				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
J2			PG6				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
K4			PG7				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
L1			PG8				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
K3			PG9				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
L2			PG10				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
L3			PG11				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
M1			PG12				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
M4			PG13				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
M2			PG14				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
N1			PG15				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
B1			PH0				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
E3			PH1				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
F4			PH2				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
B2			PH3				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
D3			PH4				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		

			User Pin				Power Supply				
Pin No.	Power Supply Name on Pin	Symbol	Function 1	Function 10	Function 11	Function 12	Name in Circuit	Permissible Input Voltage (V)	Input Buffer Type	Output Driving Ability	Notes
C3			PH5				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
V18			PJ0		TxD_B	FRTxD_A	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
V17			PJ1		RxD_B	FRRxD_A	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
V16			PJ2		TxD_B	CTx_C& CTx_D	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
U17			PJ3		RxD_B	CRx_C& CRx_D	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
T17			PJ4				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
V15			PJ5				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
W19			PJ6				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
U15			PJ7			FREN_A	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
W18			PJ8			CTx_D	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
W17			PJ9			CRx_D	$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
G2			PK0				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
H4			PK1				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
F1			PK2				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
E1			PK3				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
G3			PK4				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
F2			PK5				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
D1			PK6				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
C1			PK7				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
E2			PK8				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		
F3			PK9				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
D2			PK10				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
C2			PK11				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B	O	
P3			PL0				$PV_{cc}2$	$PV_{cc}2 + 0.3$	Schmitt B		

Note: * When higher level voltage than V_{cc} + 0.3 V is input, it is recommended that the voltage input is connected to the pins through the combined resistance of all the pins, 200 kΩ or higher.
However, although the resistance value can be lowered depending on the sink current of external V_{cc} (3-V system) regulators, the combined resistance value must be 33 kΩ at least.
When the resistance value is modified, make sure that the V_{cc} voltage does not exceed 3.6 V in the standby state.

[Usage Notes]

Set power supply voltages during LSI operation as shown below.

V_{cc} = PLLV_{cc} = 3.3 V ±0.3 V, PV_{cc}1 = 5.0 V ±0.5 V/3.3 V ±0.3 V, PV_{cc}2 = 5.0 V ±0.5 V,
AV_{cc} = 5.0 V ±0.5 V, AVREFH_A = AVREFH_B = 4.5 V to AV_{cc},
V_{ss} = PLLV_{ss} = AV_{ss} = AVREFL_A = AVREFL_B = 0 V

When PV_{cc}1 = 3.3 V ±0.3 V, V_{cc} = PV_{cc}1.

The PV_{cc}1 power supply voltage depends on the operating mode as shown below. Operation cannot be guaranteed with other PV_{cc}1 power supply voltages.

Table 34.3 PV_{CC}1 Voltage in Each Operating Mode

Operating	Pin Setting							
Mode No.	MD4	MD3	FWE	MD2	MD1	MD0	Mode Name	PV_{CC}1 Voltage
Mode 0	0	0	0*	1	1	1	MCU expansion mode	3.3 V ±0.3 V
Mode 1	0	0	1*	1	1	1		
Mode 2	0	0	0	0	0	1		
Mode 3	0	0	0	0	0	0	MCU single-chip mode	5.0 V ±0.5 V
Mode 4	0	0	1	0	1	1	Boot mode	3.3 V ±0.3 V
Mode 5	0	0	1	0	1	0		5.0 V ±0.5 V
Mode 6	0	0	1	0	0	1	User program mode	3.3 V ±0.3 V
Mode 7	0	0	1	0	0	0		5.0 V ±0.5 V
Mode 8	0	0	1	1	0	1	User boot mode	3.3 V ±0.3 V
Mode 9	0	0	1	1	0	0		5.0 V ±0.5 V

Note: * The FWE input signal determines external bus width when the on-chip ROM disabled MCU expansion mode is selected.

Tables 34.4 to 34.12 show the DC characteristics.

Table 34.4 DC Characteristics Input Level Voltage

Conditions: V_{CC} = PLLV_{CC} = 3.3 V ±0.3 V, PV_{CC}1 = 5.0 V ±0.5 V/3.3 V ±0.3 V,
PV_{CC}2 = 5.0 V ±0.5 V,
AV_{CC} = 5.0 V ±0.5 V, AVREFH_A = AVREFH_B = 4.5 V to AV_{CC},
V_{SS} = PLLV_{SS} = AV_{SS} = AVREFL_A = AVREFL_B = 0 V
When PV_{CC}1 = 3.3 V ±0.3 V, V_{CC} = PV_{CC}1.
T_a = –40°C to 125°C

Item		Symbol	Min.	Typ.	Max.	Unit	Measurement Conditions
Schmitt trigger input voltage (buffer type A for mode and control pins)	$\overline{\text{HSTBY}}$, $\overline{\text{RES}}$, NMI, FWE, MD2 to MD0	V_T^+ (V_{IH})	V_{CC} × 0.83	—	5.5 + 0.3*[2]	V	Refer to table 34.2 (pins with Schmitt A input buffers)
	ASEMD, MD4, MD3, MD_CLK1, MD_CLK0, MD_CLKP	V_T^+ (V_{IH})	V_{CC} × 0.83	—	V_{CC} + 0.3*[3]	V	
		V_T^+ (V_{IH})	V_{CC} × 0.83	—	5.5 + 0.3*[1]	V	
	$\overline{\text{HSTBY}}$, $\overline{\text{RES}}$, NMI, ASEMD, FWE, MD4 to MD0, MD_CLK1, MD_CLK0, MD_CLKP	V_T^- (V_{IL})	–0.3	—	V_{CC} × 0.2	V	
		V_{HS}	V_{CC} × 0.15	—	—	V	
	$\overline{\text{TRST}}$, AUDMD, $\overline{\text{AUDRST}}$	V_T^+ (V_{IH})	V_{CC} × 0.83	—	V_{CC} + 0.3	V	
		V_T^- (V_{IL})	- 0.3	—	V_{CC} × 0.2	V	
		V_{HS}	V_{CC} × 0.15	—	—	V	
Schmitt trigger input voltage (buffer type B for GPIO and peripheral IO pins)	PA15 to PA0, PB14 to PB0, PD13 to PD0, PE13 to PE0, PF15 to PF0, PG15 to PG0, PH5 to PH0, PJ9 to PJ0, PK11 to PK0, PL8 to PL0	V_T^+ (V_{IH})	PV_{CC} × 0.7	—	PV_{CC} + 0.3	V	Refer to table 34.2 (pins with Schmitt B input buffers)
		V_T^- (V_{IL})	–0.3	—	PV_{CC} × 0.42	V	
		V_{HS}	PV_{CC} × 0.082	—	—	V	
TTL input voltage (for GPIO pins)	PC15 to PC0	V_{IH}	PV_{CC}1 × 0.7	—	PV_{CC}1 + 0.3	V	PV_{CC}1 = PV_{CC}2 = 5.0 V ±0.5 V
		V_{IL}	–0.3	—	PV_{CC}1 × 0.3	V	

Item		Symbol	Min.	Typ.	Max.	Unit	Measurement Conditions
TTL input voltage (for extended data bus pins)	D15 to D0 (MCU expansion mode)	V_{IH}	2.2	—	$PV_{CC}1 + 0.3$	V	$PV_{CC}1 = V_{CC} = 3.3\ V \pm 0.3\ V$
		V_{IL}	–0.3	—	0.8	V	
Clock input pin voltage	EXTAL	V_{IH}	$V_{CC} \times 0.7$	—	$V_{CC} + 0.3$	V	
		V_{IL}	–0.3	—	$V_{CC} \times 0.2$	V	
TTL input voltage (for AUD-II and H-UDI)	AUDCK, $\overline{\text{AUDSYNC}}$, AUDATA3 to AUDATA0, TCK, TMS, TDI	V_{IH}	2.2	—	$V_{CC} + 0.3$	V	
		V_{IL}	–0.3	—	0.8	V	
Schmitt trigger input voltage (buffer type B for JTAG)	TMS, TDI	V_T^+ (V_{IH})	$V_{CC} \times 0.7$	—	$V_{CC} + 0.3$	V	Refer to table 34.2 (pins with Schmitt B input buffers)
		V_T^- (V_{IL})	–0.3	—	$V_{CC} \times 0.42$	V	
		V_{HS}	$V_{CC} \times 0.082$	—	—	V	

Notes: 1. When higher level voltage than V_{CC} + 0.3 V is input, it is recommended that the voltage input is connected to the pins through the combined resistance of all the pins, 200 kΩ or higher.
However, although the resistance value can be lowered depending on the sink current of external V_{CC} (3-V system) regulators, the combined resistance value must be 33 kΩ at least.
When the resistance value is modified, make sure that the V_{CC} voltage does not exceed 3.6 V in the standby state.

2. When the ASEMD, MD4, MD3, MD_CLK1, MD_CLK0, and MD_CLKP pins are input in 5-V tolerant, it is recommended that the $\overline{\text{HSTBY}}$, $\overline{\text{RES}}$, NMI, FWE, MD0, MD1, and MD2 pins are input in 5-V amplitude.

3. When the $\overline{\text{ASEMD}}$, $\overline{\text{MD4}}$, MD3, MD_CLK1, MD_CLK0, and MD_CLKP pins are input in 3.3-V, the $\overline{\text{HSTBY}}$, $\overline{\text{RES}}$, NMI, FWE, MD0, MD1, and MD2 pins can be input in either 3.3-V or 5-V amplitude.

Table 34.5 DC Characteristics Input Leak Current

Conditions: V_{CC} = PLLV_{CC} = 3.3 V ±0.3 V, PV_{CC}1 = 5.0 V ±0.5 V/3.3 V ±0.3 V,
PV_{CC}2 = 5.0 V ±0.5 V,
AV_{CC} = 5.0 V ±0.5 V, AVREFH_A = AVREFH_B = 4.5 V to AV_{CC},
V_{SS} = PLLV_{SS} = AV_{SS} = AVREFL_A = AVREFL_B = 0 V
When PV_{CC}1 = 3.3 V ±0.3 V, V_{CC} = PV_{CC}1.
T_a = –40°C to 125°C

Item		Symbol	Min.	Typ.	Max.	Unit	Measurement Conditions
Input leak current	$\overline{\text{HSTBY}}$, $\overline{\text{RES}}$, NMI, ASEMD, FWE, MD4 to MD0, MD_CLK1, MD_CLK0, MD_CLKP, EXTAL	| lin |	—	—	2.0	μA	Vin = 0.3 V to V_{CC} – 0.3 V
	$\overline{\text{TRST}}$, TMS, TDI, TCK, AUDMD, $\overline{\text{AUDRST}}$, AUDCK, $\overline{\text{AUDSYNC}}$, AUDATA3 to AUDATA0 (when pull-up/down resistor is off)		—	—	2.0	μA	Vin = 0.3 V to V_{CC} – 0.3 V
	PA15 to PA0, PB14 to PB0 (extended bus mode)		—	—	2.0	μA	Vin = 0.3 V to PV_{CC}1 – 0.3 V, PV_{CC}1 = V_{CC} = 3.3 V ± 0.3 V
	PC15 to PC0 (D15 to D0) (standby in expansion bus mode)		—	—	2.0	μA	Vin = 0.3 V to PV_{CC}1 – 0.3 V, PV_{CC}1 = V_{CC} = 3.3 V ± 0.3 V
	PA15 to PA0, PB14 to PB0, PC15 to PC0 (other than extended bus mode)		—	—	2.0	μA	Vin = 0.3 V to PV_{CC}1 – 0.3 V, PV_{CC}1 = PV_{CC}2 = 5.0 V ± 0.5 V
	PD13 to PD0, PE13 to PE0, PF15 to PF0, PG15 to PG0, PH5 to PH0, PJ9 to PJ0, PK11 to PK0, PL8 to PL0		—	—	2.0	μA	Vin = 0.3 V to PV_{CC}2 – 0.3 V

Item		Symbol	Min.	Typ.	Max.	Unit	Measurement Conditions
Input leak current	A/D port	| Iin |	—	—	0.1	μA	Vin = 0.3 V to AV_{CC} – 0.3 V, T_a = –40°C to 105°C
			—	—	0.2	μA	Vin = 0.3 V to AV_{CC} – 0.3 V, T_a = 105°C to 125°C

Table 34.6 DC Characteristics Pull-Up/Pull-Down MOS Current

Conditions: V_{CC} = $PLLV_{CC}$ = 3.3 V ±0.3 V, $PV_{CC}1$ = 5.0 V ±0.5 V/3.3 V ±0.3 V,
$PV_{CC}2$ = 5.0 V ±0.5 V,
AV_{CC} = 5.0 V ±0.5 V, AVREFH_A = AVREFH_B = 4.5 V to AV_{CC},
V_{SS} = $PLLV_{SS}$ = AV_{SS} = AVREFL_A = AVREFL_B = 0 V
When $PV_{CC}1$ = 3.3 V ±0.3 V, V_{CC} = $PV_{CC}1$.
T_a = –40°C to 125°C

Item		Symbol	Min.	Typ.	Max.	Unit	Measurement Conditions
Input pull-up MOS current	TMS, $\overline{\text{TRST}}$, TDI, TCK	–Ipu	—	—	350	μA	Vin = 0 V
	AUDMD, AUDCK, $\overline{\text{AUDSYNC}}$, AUDATA3 to AUDATA0		—	—	350	μA	Vin = 0 V
	D15 to D0 (MCU expansion mode)		—	—	160	μA	Vin = 0 V
Input pull-down MOS current	$\overline{\text{AUDRST}}$	Ipd	—	—	350	μA	Vin = V_{CC}
	RxD_A to RxD_E		—	—	350	μA	Vin = PV_{CC}
	MISOA to MISOC		—	—	350	μA	Vin = PV_{CC}
	ASEMD		—	—	200	μA	Vin = V_{CC}

Table 34.7 DC Characteristics — Output Level Voltage

Conditions: $V_{CC} = PLLV_{CC}$ = 3.3 V ±0.3 V, $PV_{CC}1$ = 5.0 V ±0.5 V/3.3 V ±0.3 V,
$PV_{CC}2$ = 5.0 V ±0.5 V,
AV_{CC} = 5.0 V ±0.5 V, AVREFH_A = AVREFH_B = 4.5 V to AV_{CC},
$V_{SS} = PLLV_{SS} = AV_{SS}$ = AVREFL_A = AVREFL_B = 0 V,
When $PV_{CC}1$ = 3.3 V ±0.3 V, $V_{CC} = PV_{CC}1$.
T_a = –40°C to 125°C

Item		Symbol	Min.	Typ.	Max.	Unit	Measurement Conditions
Output high-level voltage	PA15 to PA0, PB14, PB13, PB11 to PB1, PC15 to PC0 (MCU expansion mode)	V_{OH}	$PV_{CC}1$ – 0.5	—	—	V	I_{OH} = 200 µA, $PV_{CC}1 = V_{CC}$ = 3.3 V ± 0.3 V
	PA15 to PA0, PB14, PB13, PB11 to PB1, PC15 to PC0 (When not in MCU expansion mode)		$PV_{CC}1$ – 0.5	—	—	V	I_{OH} = 200 µA, $PV_{CC}1 = PV_{CC}2$ = 5.0 V ± 0.5 V
			$PV_{CC}1$ – 1.0	—	—	V	I_{OH} = 1 mA, $PV_{CC}1 = PV_{CC}2$ = 5.0 V ± 0.5 V
	PD13 to PD0, PE13 to PE0, PF15, PF13 to PF0, PG15 to PG0, PH5 to PH0, PJ9, PJ8, PJ6 to PJ1, PK11 to PK0, PL8 to PL0, PJ0, PB0, PF14, PJ7, PB12		$PV_{CC}2$ – 0.5	—	—	V	I_{OH} = 200 µA
			$PV_{CC}2$ – 1.0	—	—	V	I_{OH} = 1 mA
	CK		V_{CC} – 0.5	—	—	V	I_{OH} = 200 µA
	$\overline{\text{WDTOVF}}$		V_{CC} – 0.5	—	—	V	I_{OH} = 200 µA
	TDO		V_{CC} – 0.5	—	—	V	I_{OH} = 200 µA
	AUDCK, $\overline{\text{AUDSYNC}}$, AUDATA3 to AUDATA0		V_{CC} – 0.5	—	—	V	I_{OH} = 200 µA

Item		Symbol	Min.	Typ.	Max.	Unit	Measurement Conditions
Output low-level voltage	PA15 to PA0, PB14, PB13, PB11 to PB1, PC15 to PC0 (MCU expansion mode)	V_{OL}	—	—	0.4	V	I_{OL} = 1.6 mA, $PV_{CC}1 = V_{CC}$ = 3.3 V ± 0.3 V
	PA15 to PA0, PB14, PB13, PB11 to PB1, PC15 to PC0 (other than MCU expansion mode)		—	—	0.4	V	I_{OL} = 1.6 mA, $PV_{CC}1 = PV_{CC}2$ = 5.0 V ± 0.5 V
			—	—	1.2	V	I_{OL} = 4 mA, $PV_{CC}1 = PV_{CC}2$ = 5.0 V ± 0.5 V
	PD13 to PD0, PE13 to PE0, PF15, PF13 to PF0, PG15 to PG0, PH5 to PH0, PJ9, PJ8, PJ6 to PJ1, PK11 to PK0, PL8 to PL0, PJ0, PB0, PF14, PJ7, PB12		—	—	0.4	V	I_{OL} = 1.6 mA
			—	—	1.2	V	I_{OL} = 4 mA
	CK		—	—	0.4	V	I_{OL} = 1.6 mA
	$\overline{\text{WDTOVF}}$		—	—	0.4	V	I_{OL} = 1 mA
	TDO		—	—	0.4	V	I_{OL} = 1.6 mA
	AUDCK, $\overline{\text{AUDSYNC}}$, AUDATA3 to AUDATA0		—	—	0.4	V	I_{OL} = 1.6 mA

Table 34.8 DC Characteristics — Permissible Output Current Values

Conditions: V_{CC} = PLLV_{CC} = 3.3 V ±0.3 V, PV_{CC}1 = 5.0 V ±0.5 V/3.3 V ±0.3 V,
PV_{CC}2 = 5.0 V ±0.5 V,
AV_{CC} = 5.0 V ±0.5 V, AVREFH_A = AVREFH_B = 4.5 V to AV_{CC},
V_{SS} = PLLV_{SS} = AV_{SS} = AVREFL_A = AVREFL_B = 0 V
When PV_{CC}1 = 3.3 V ±0.3 V, V_{CC} = PV_{CC}1.
T_a = –40°C to 125°C

Item	Symbol	Min.	Typ.	Max.	Unit
Output low-level permissible current (per pin)	I_{OL}	—	—	4.0	mA
Output low-level permissible current (total)	ΣI_{OL}	—	—	80	mA
Output high-level permissible current (per pin)	I_{OH}	—	—	2.0	mA
Output high-level permissible current (total)	ΣI_{OH}	—	—	25	mA

[Operating precautions]

To assure LSI reliability, do not exceed the output values listed in this table.

Table 34.9 DC Characteristics — Injection Current Values

Conditions: V_{CC} = PLLV_{CC} = 3.3 V ±0.3 V, PV_{CC}1 = 5.0 V ±0.5 V/3.3 V ±0.3 V,
PV_{CC}2 = 5.0 V ±0.5 V,
AV_{CC} = 5.0 V ±0.5 V, AVREFH_A = AVREFH_B = 4.5 V to AV_{CC},
V_{SS} = PLLV_{SS} = AV_{SS} = AVREFL_A = AVREFL_B = 0 V
When PV_{CC}1 = 3.3 V ±0.3 V, V_{CC} = PV_{CC}1.
T_a = –40°C to 125°C

Item		Symbol	Min.	Typ.	Max.	Unit
DC Injection current (per pin)	logic pin	I_{IC}	–1.0	—	2.0	mA
	Analog pin		–3.0	—	3.0	mA
DC Injection current (total)		$\Sigma \mid I_{IC} \mid$	—	—	50.0	mA

Note: Make sure that the voltages for the pins do not exceed 5.8 V.

Table 34.10 DC Characteristics — Input Capacitance

Conditions: V_{CC} = PLLV_{CC} = 3.3 V ±0.3 V, PV_{CC}1 = 5.0 V ±0.5 V/3.3 V ±0.3 V,
PV_{CC}2 = 5.0 V ±0.5 V,
AV_{CC} = 5.0 V ±0.5 V, AVREFH_A = AVREFH_B = 4.5 V to AV_{CC},
V_{SS} = PLLV_{SS} = AV_{SS} = AVREFL_A = AVREFL_B = 0 V
When PV_{CC}1 = 3.3 V ±0.3 V, V_{CC} = PV_{CC}1.

Item		Symbol	Min.	Typ.	Max.	Unit	Measurement Conditions
Input capacitance	All input pins	Cin	—	10	20	pF	Vin = 0 V, f = 1 MHz, T_a = 25°C

Table 34.11 DC Characteristics — Supply Current

Conditions: V_{CC} = PLLV_{CC} = 3.3 V ±0.3 V, PV_{CC}1 = 5.0 V ±0.5 V/3.3 V ±0.3 V,
PV_{CC}2 = 5.0 V ±0.5 V,
AV_{CC} = 5.0 V ±0.5 V, AVREFH_A = AVREFH_B = 4.5 V to AV_{CC},
V_{SS} = PLLV_{SS} = AV_{SS} = AVREFL_A = AVREFL_B = 0 V
When PV_{CC}1 = 3.3 V ±0.3 V, V_{CC} = PV_{CC}1.
T_a = –40°C to 125°C

Item		Symbol	Min.	Typ.*	Max.	Unit	Measurement Conditions
Supply current (V_{CC} power supply)	Normal operation (Including flash memory programming and erasure)	I_{CC}	—	104	175	mA	f = 80 MHz
			—	127	198	mA	f = 120 MHz
			—	150	221	mA	f = 160 MHz
			—	174	244	mA	f = 200 MHz
	Sleep	I_{SLP}	—	53	112	mA	f = 80 MHz
			—	59	118	mA	f = 120 MHz
			—	65	124	mA	f = 160 MHz
			—	70	130	mA	f = 200 MHz
	At reset	I_{RST}	30	66	129	mA	f = 80 MHz
			40	75	138	mA	f = 120 MHz
			50	85	148	mA	f = 160 MHz
			60	94	157	mA	f = 200 MHz
PLL supply current (PLLV_{CC} power supply)		I_{PLL}	—	4.0	7.5	mA	
Analog supply current (AV_{CC} power supply)	During A/D conversion	I_{AVCC}	—	7.4	9.8	mA	for 2 modules
	Awaiting A/D conversion		—	65	120	µA	
	Standby		—	5.0	100	µA	
ADC reference power supply current (AVREF)	During A/D conversions	I_{AVREF}	—	2.5	4.0	mA	for 2 modules
	Awaiting A/D conversion		—	2.2	3.5	mA	
	Standby		—	0.1	1.0	µA	

Note: * The Typ condition is V_{CC} = 3.3V, T_a = 25 °C.

[Operating precautions]

1. When the A/D converter is not used (including during standby), do not leave the AV_{CC}, AV_{ref}, and AV_{SS} pins open.
2. The supply current is measured when V_{IH}min = V_{CC} – 0.5 V/PV_{CC} – 0.5 V, V_{IL} = 0.5 V, with all output pins unloaded.
3. The guaranteed operating range of power supply $PV_{CC}1$ in MCU expansion modes is only $PV_{CC}1$ = 3.3 V ±0.3 V. Do not use a voltage outside this range.
4. The guaranteed operating range of power supply $PV_{CC}1$ in MCU single-chip mode is only $PV_{CC}1$ = 5.0 V ±0.5 V. Do not use a voltage outside this range.

Table 34.12 DC Characteristics — Standby

Conditions: V_{CC} = $PLLV_{CC}$ = 3.3 V ±0.3 V, $PV_{CC}1$ = 5.0 V ±0.5 V/3.3 V ±0.3 V,
$PV_{CC}2$ = 5.0 V ±0.5 V,
AV_{CC} = 5.0 V ±0.5 V, AVREFH_A = AVREFH_B = 4.5 V to AV_{CC},
V_{SS} = $PLLV_{SS}$ = AV_{SS} = AVREFL_A = AVREFL_B = 0 V,
When $PV_{CC}1$ = 3.3 V ±0.3 V, V_{CC} = $PV_{CC}1$.
T_a = –40°C to 125°C

Item		Symbol	Min.	Typ.	Max.	Unit	Measurement Conditions
Supply current (V_{CC} power supply)	Standby	I_{SB}	—	—	300	μA	-40°C ≤ Ta ≤ 50°C
			—	—	750	μA	50°C < Ta ≤ 105°C
			—	—	1000	μA	105°C < Ta ≤ 125°C
RAM standby voltage (V_{CC} power supply)	RAM data retention	V_{RAM}	2.7	—	—	V	

34.3 AC Characteristics

34.3.1 Timing for Power On and Off

Table 34.13 shows the timing for power on and off.

Table 34.13 Timing for Power On and Off

Conditions: V_{CC} = PLLV_{CC} = 3.3 V ±0.3 V, PV_{CC}1 = 5.0 V ±0.5 V/3.3 V ±0.3 V,
PV_{CC}2 = 5.0 V ±0.5 V,
AV_{CC} = 5.0 V ±0.5 V, AVREFH_A = AVREFH_B = 4.5 V to AV_{CC},
V_{SS} = PLLV_{SS} = AV_{SS} = AVREFL_A = AVREFL_B = 0 V
When PV_{CC}1 = 3.3 V ±0.3 V, V_{CC} = PV_{CC}1.
T_a = –40°C to 125°C

Item	Symbol	Min.	Max.	Unit	Figures
Preceding Vcc power-on time	t_{VCCS}	0	—	ms	Figure 34.1
Vcc holding time at PVcc shutdown	t_{VCCH1}	0	—	ms	

Note: If t_{VCCS} or t_{VCCH1} are not satisfied, status of pins operating on PVcc1 or PVcc2 cannot be guaranteed.

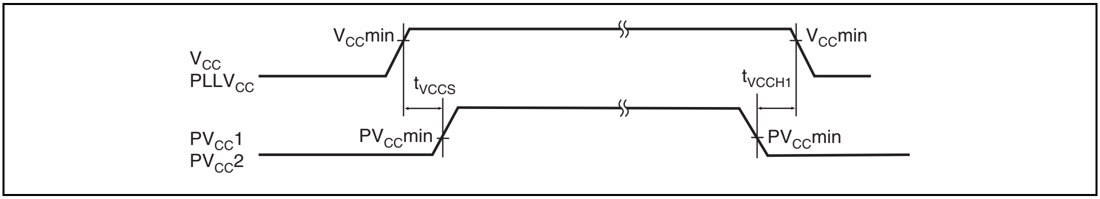


Figure 34.1 Timing for Power On and Off

34.3.2 Timing for Operation Mode and Oscillation

Table 34.14 shows the timing for operation mode and oscillation.

Table 34.14 Timing for operation mode and oscillation

Conditions: V_{CC} = PLLV_{CC} = 3.3 V ±0.3 V, PV_{CC}1 = 5.0 V ±0.5 V/3.3 V ±0.3 V,
PV_{CC}2 = 5.0 V ±0.5 V,
AV_{CC} = 5.0 V ±0.5 V, AVREFH_A = AVREFH_B = 4.5 V to AV_{CC},
V_{SS} = PLLV_{SS} = AV_{SS} = AVREFL_A = AVREFL_B = 0 V
When PV_{CC}1 = 3.3 V ±0.3 V, V_{CC} = PV_{CC}1.
T_a = –40°C to 125°C

Item		Symbol	Min.	Max.	Unit	Figures
Oscillation settling time		t_{OSC1}	10	—	ms	Figure 34.2
Operation mode set up time when start-up		t_{MDS1}	10	—	ms	
Operation mode set up time during operation	MD_CLKP = 0	t_{MDS2}	10	—	t_{cyc}	
	MD_CLKP = 1		20	—		
Operation mode hold time after reset is inactive	MD_CLKP = 0	t_{MDH1}	30	—	t_{cyc}	
	MD_CLKP = 1		60	—		
Operation mode hold time when power down		t_{MDH2}	0	—	ms	
V_{CC} hold time when switched off. (Avoid damage during flash programming or erasing)		t_{VCCH2}	22	—	µs	
$\overline{\text{HSTBY}}$ hold time after reset (Keep RAM contents during reset state)		t_{HSTBYH}	1	—	µs	
$\overline{\text{HSTBY}}$ pulse width (Power down and re-start internal regulator)		t_{HSTBYW}	1	—	ms	Figure 34.3

Note: AVREFH_A ≤ AVcc + 0.3 V and AVREFH_B ≤ AVcc + 0.3 V must always be satisfied even at power-on/off.

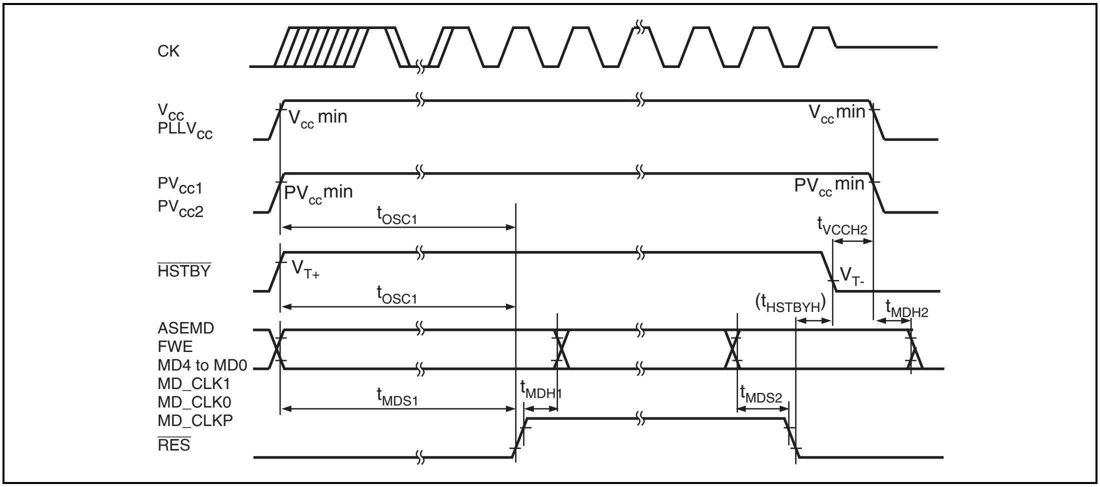


Figure 34.2 Operation Mode and Oscillation Timing when Power-On/Off

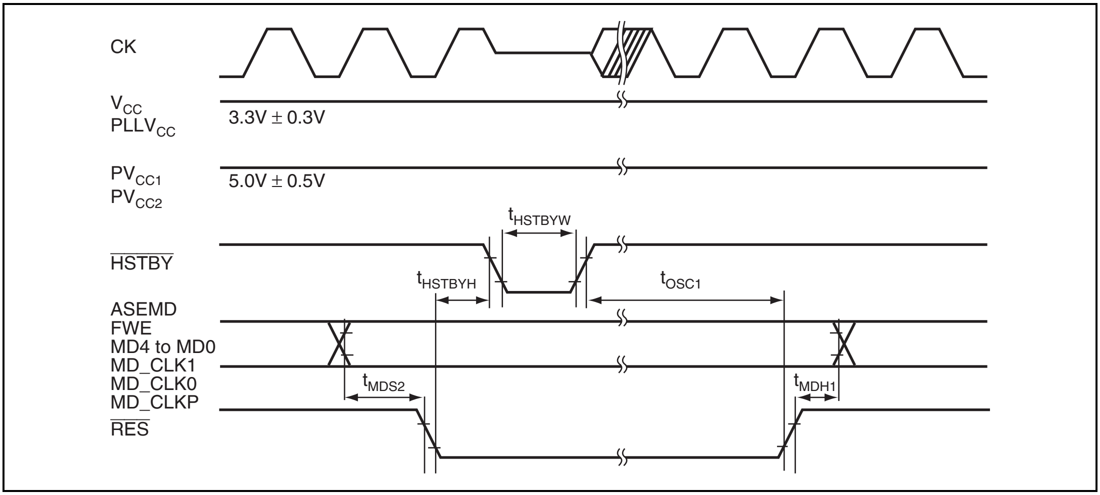


Figure 34.3 Operation Mode and Oscillation Timing during Operation

34.3.3 Clock Timing

Table 34.15 shows the clock timing.

Table 34.15 Clock Timing

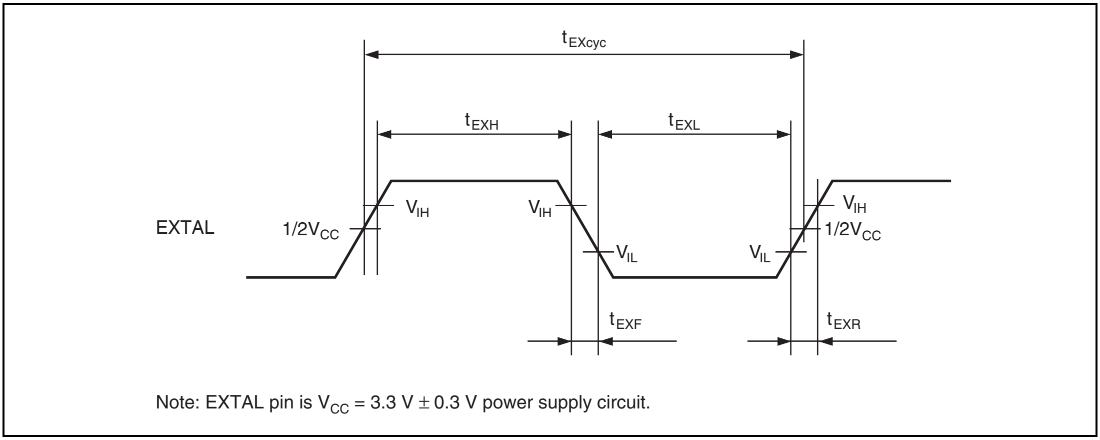
Conditions: V_{CC} = PLLV_{CC} = 3.3 V ±0.3 V, PV_{CC}1 = 5.0 V ±0.5 V/3.3 V ±0.3 V,
PV_{CC}2 = 5.0 V ±0.5 V,
AV_{CC} = 5.0 V ±0.5 V, AVREFH_A = AVREFH_B = 4.5 V to AV_{CC},
V_{SS} = PLLV_{SS} = AV_{SS} = AVREFL_A = AVREFL_B = 0 V
When PV_{CC}1 = 3.3 V ±0.3 V, V_{CC} = PV_{CC}1.
T_a = –40°C to 125°C

Item		Symbol	Min.	Max.	Unit	Figures
EXTAL clock input frequency		f_{EX}	16	20	MHz	Figure 34.4
EXTAL clock input cycle time		t_{EXcyc}	50	62.5	ns	
EXTAL clock input low-level pulse width		t_{EXL}	15	—	ns	
EXTAL clock input high-level pulse width		t_{EXH}	15	—	ns	
EXTAL clock input rise time		t_{EXR}	—	4	ns	
EXTAL clock input fall time		t_{EXF}	—	4	ns	
Clock frequency*[1]	MD_CLKP = 0	f_{op}	16	20	MHz	Figure 34.5
Clock cycle time		t_{cyc}	50	62.5	ns	
Clock low-level pulse width		t_{CL}	12	—	ns	
Clock high-level pulse width		t_{CH}	12	—	ns	
Clock frequency*[1]	MD_CLKP = 1	f_{op}	32	40	MHz	
Clock cycle time		t_{cyc}	25	31.25	ns	
Clock low-level pulse width		t_{CL}	4	—	ns	
Clock high-level pulse width		t_{CH}	4	—	ns	
Clock rise time		t_{CR}	—	8	ns	
Clock fall time		t_{CF}	—	8	ns	
Clock jitter (averaged 1µs)*[2]		f_{JIT}	—	0.3	%	Not tested

Notes: 1. The CK pin outputs the peripheral clock signal (Pϕ).
2. The clock jitter depends on board design.

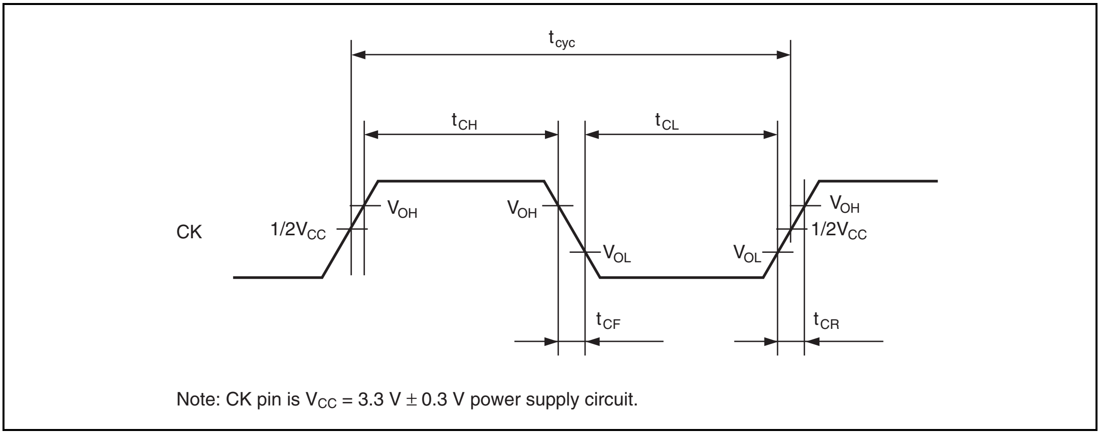
[Operating precautions]

The EXTAL, XTAL, and CK pins constitute a circuit requiring a power supply voltage of V_{CC} = 3.3 V ±0.3 V. Comply with the input and output voltages specified in the DC characteristics.



Note: EXTAL pin is V_{CC} = 3.3 V ± 0.3 V power supply circuit.

Figure 34.4 EXTAL Clock Input Timing



Note: CK pin is V_{CC} = 3.3 V ± 0.3 V power supply circuit.

Figure 34.5 Peripheral Clock Timing

34.3.4 Control Signal Timing

Table 34.16 shows the control signal timing.

Table 34.16 Control Signal Timing

Conditions: V_{CC} = PLLV_{CC} = 3.3 V ±0.3 V, PV_{CC}1 = 5.0 V ±0.5 V/3.3 V ±0.3 V,
PV_{CC}2 = 5.0 V ±0.5 V,
AV_{CC} = 5.0 V ±0.5 V, AVREFH_A = AVREFH_B = 4.5 V to AV_{CC},
V_{SS} = PLLV_{SS} = AV_{SS} = AVREFL_A = AVREFL_B = 0 V
When PV_{CC}1 = 3.3 V ±0.3 V, V_{CC} = PV_{CC}1.
T_a = –40°C to 125°C

Item		Symbol	Min.	Max.	Unit	Figures
$\overline{\text{RES}}$ pulse width (Except flash programming or erasing)	MD_CLKP = 0	t_{RESW1}	30	—	t_{cyc}	Figure 34.6
	MD_CLKP = 1		60	—		
$\overline{\text{RES}}$ pulse width (During flash programming or erasing)*		t_{RESW2}	20	—	µs	
$\overline{\text{RES}}$ noise canceling width		t_{RESNCW}	0.2	1.3	µs	
$\overline{\text{RES}}$ setup time		t_{RESS}	1.3	—	µs	
NMI setup time		t_{NMIS}	330	—	ns	Figure 34.7
NMI hold time		t_{NMIH}	330	—	ns	
$\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$ setup time		t_{IRQS}	24	—	ns	
$\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$ hold time (edge detection)		t_{IRQH}	24	—	ns	

Note: * Wait time to discharge internal high voltage for flash programming/erasing.

[Operating precautions]

1. Mode setup time during power-on reset by the $\overline{\text{RES}}$ pin depends on the combination of signals to be input to the FWE, MD4 to MD0, MD_CLK1 to MD_CLK0, and MD_CLKP pins. If a low-level signal is input to the $\overline{\text{RES}}$ pin while this LSI operates by inputting a mode specified in table 34.3 to the FWE, MD4 to MD0, MD_CLK1 to MD_CLK0, and MD_CLK pins, the mode setup time is defined by t_{MDS2}. If a signal other than the combination of signals specified in table 34.3 (undefined mode) is input to the FWE, MD4 to MD0, MD_CLK1, MD_CLK0, and MD_CLKP pins, the mode setup time is defined by t_{MDS1}.
2. The $\overline{\text{RES}}$, NMI, and $\overline{\text{IRQ7}}$–$\overline{\text{IRQ0}}$ signals are asynchronous inputs, but when the setup times shown here are provided, the signals are considered to have been changed at clock rise. If the setup times are not provided, recognition is delayed until the next clock rise.

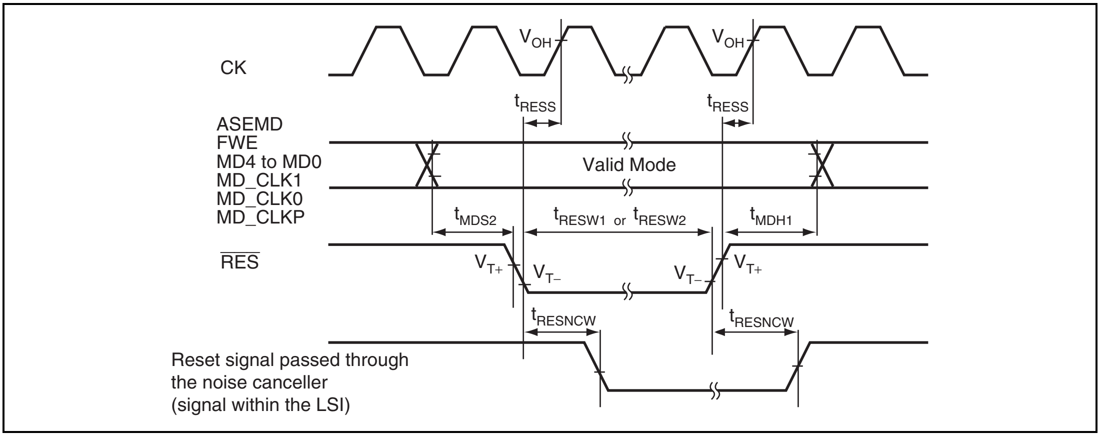


Figure 34.6 Reset Input Timing

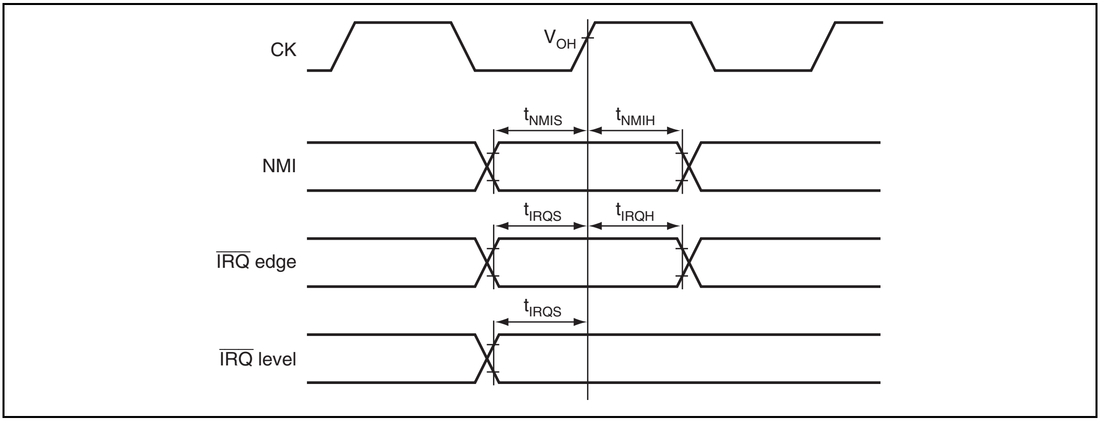


Figure 34.7 Interrupt Signal Input Timing

34.3.5 Bus Timing

Table 34.17 shows the bus timing.

Table 34.17 Bus Timing

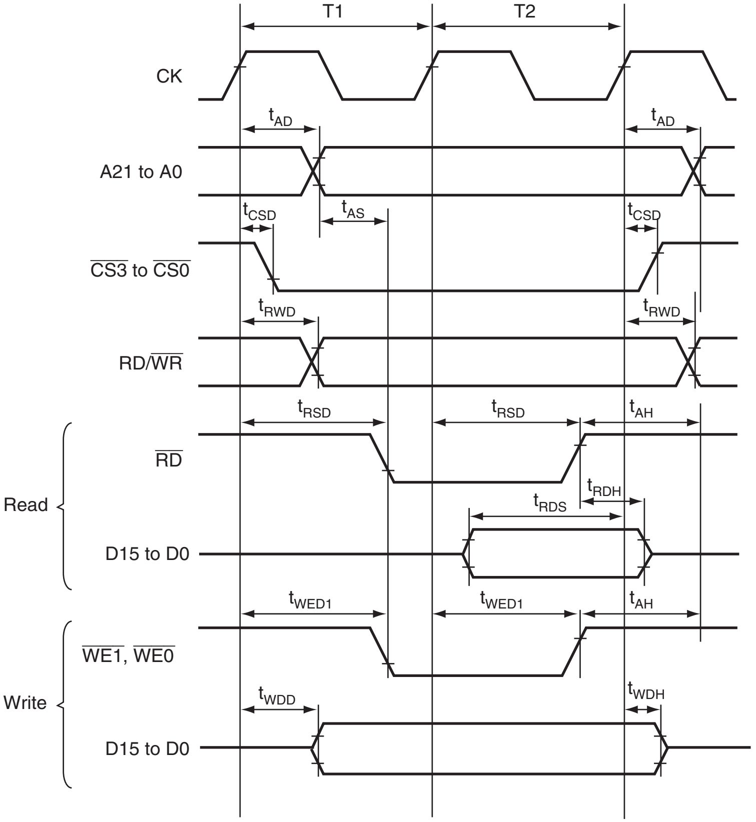
Conditions: V_{CC} = $PLLV_{CC}$ = 3.3 V ±0.3 V, $PV_{CC}1$ = 5.0 V ±0.5 V/3.3 V ±0.3 V,
$PV_{CC}2$ = 5.0 V ±0.5 V,
AV_{CC} = 5.0 V ±0.5 V, AVREFH_A = AVREFH_B = 4.5 V to AV_{CC},
V_{SS} = $PLLV_{SS}$ = AV_{SS} = AVREFL_A = AVREFL_B = 0 V
When $PV_{CC}1$ = 3.3 V ±0.3 V, V_{CC} = $PV_{CC}1$.
T_a = –40°C to 125°C

Item	Symbol	Min.	Max.	Unit	Figures
Address delay time	t_{AD}	—	35	ns	Figures 34.8 to 34.13
Address setup time	t_{AS}	0	—	ns	Figures 34.8 to 34.11
Address hold time	t_{AH}	0	—	ns	
$\overline{\text{CS}}$ delay time	t_{CSD}	—	30	ns	Figures 34.8 to 34.13
Read/write delay time	t_{RWD}	—	30	ns	
Read strobe delay time	t_{RSD}	$1/2t_{cyc}$	$1/2t_{cyc}$ + 30	ns	
Read data setup time	t_{RDS}	$1/2t_{cyc}$ + 15	—	ns	
Read data hold time	t_{RDH}	0	—	ns	
Write enable delay time 1	t_{WED1}	$1/2t_{cyc}$	$1/2t_{cyc}$ + 30	ns	Figures 34.8 to 34.12
Write enable delay time 2	t_{WED2}	—	30	ns	Figure 34.13
Write data delay time	t_{WDD}	—	30	ns	Figures 34.8 to 34.13
Write data hold time	t_{WDH}	0	—	ns	
$\overline{\text{WAIT}}$ setup time	t_{WTS}	$1/2t_{cyc}$ + 15	—	ns	Figures 34.9 to 34.13
$\overline{\text{WAIT}}$ hold time	t_{WTH}	$1/2t_{cyc}$ + 0	—	ns	

[Operating precautions]

The guaranteed operating range of power supply $PV_{CC}1$ in the MCU expanded modes is only $PV_{CC}1$ = 3.3 V ± 0.3 V. Do not use a voltage outside this range.

Note: $1/2t_{cyc}$ for the delay, setup, and hold time in the above table indicates the falling edge of the clock signal.



Note: t_{RDS} indicates read data setup time. Valid read data should not necessarily be held till the rising edge of CK at the end of T2 unless the AC rules for t_{RSD} and t_{RDH} are violated.

Figure 34.8 Bus Timing (No Waits)

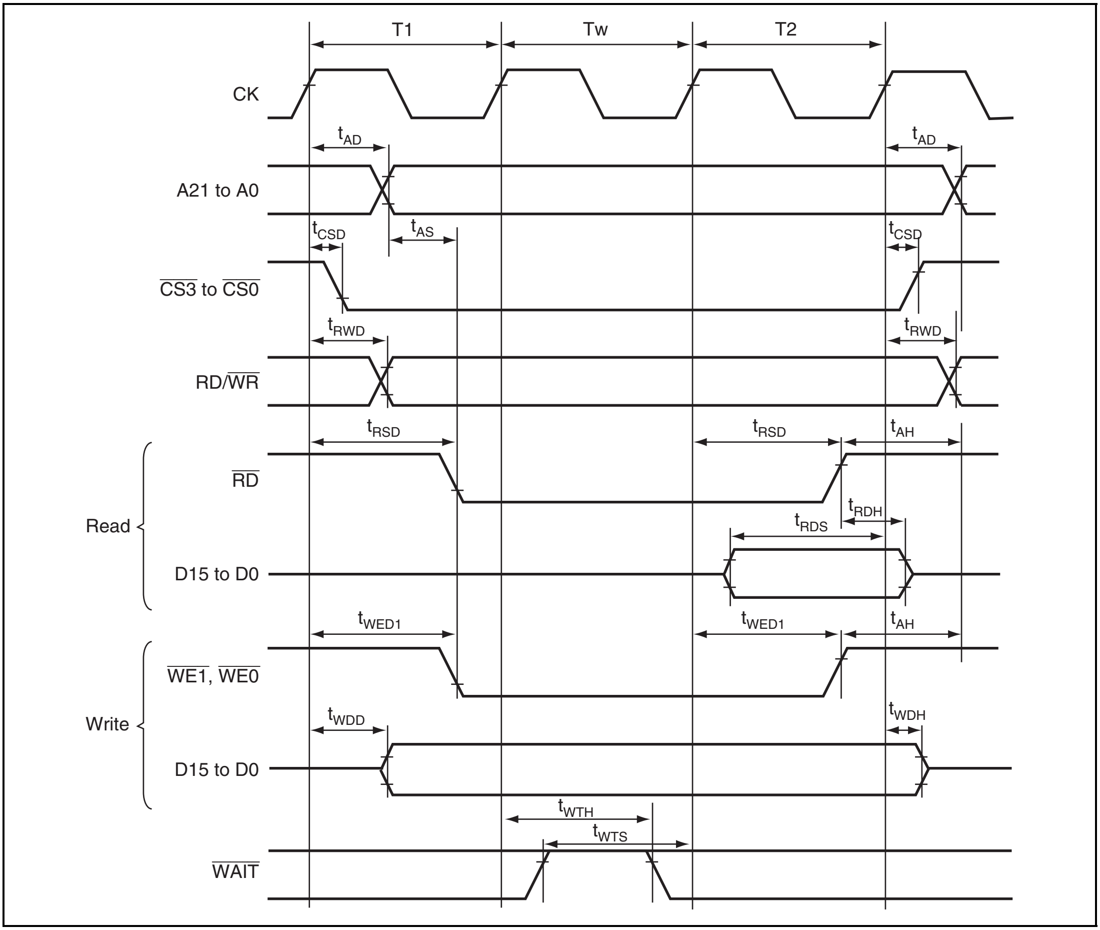


Figure 34.9 Bus Timing (One Software Wait Cycle)

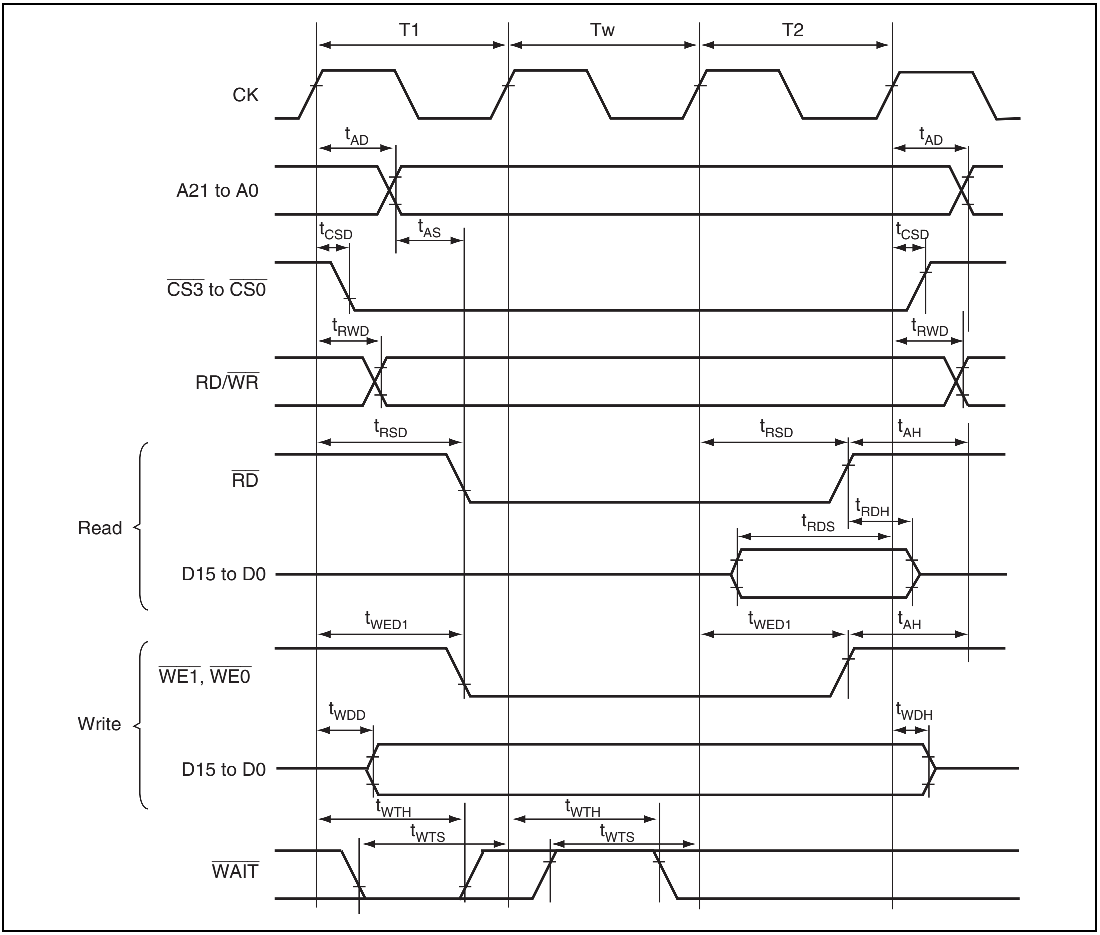


Figure 34.10 Bus Timing (One External Wait Cycle)

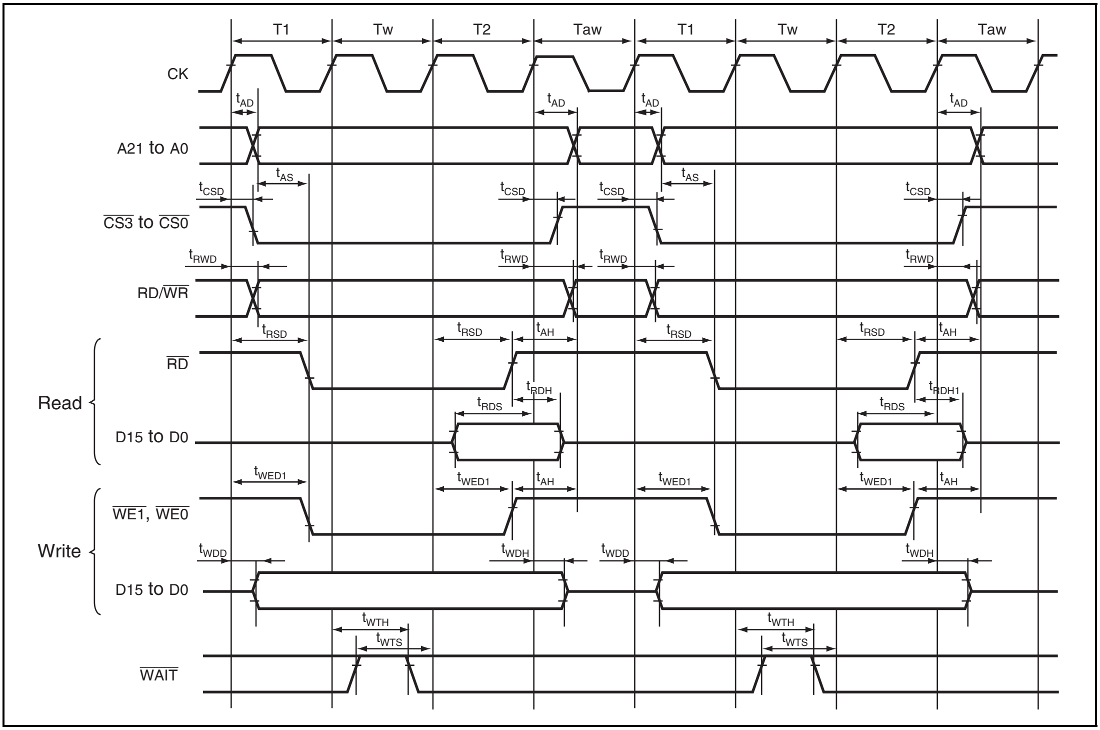


Figure 34.11 Bus Timing
(One Software Wait Cycle, One External Wait Cycle Valid (WM Bit = 0), No Idle Cycle)

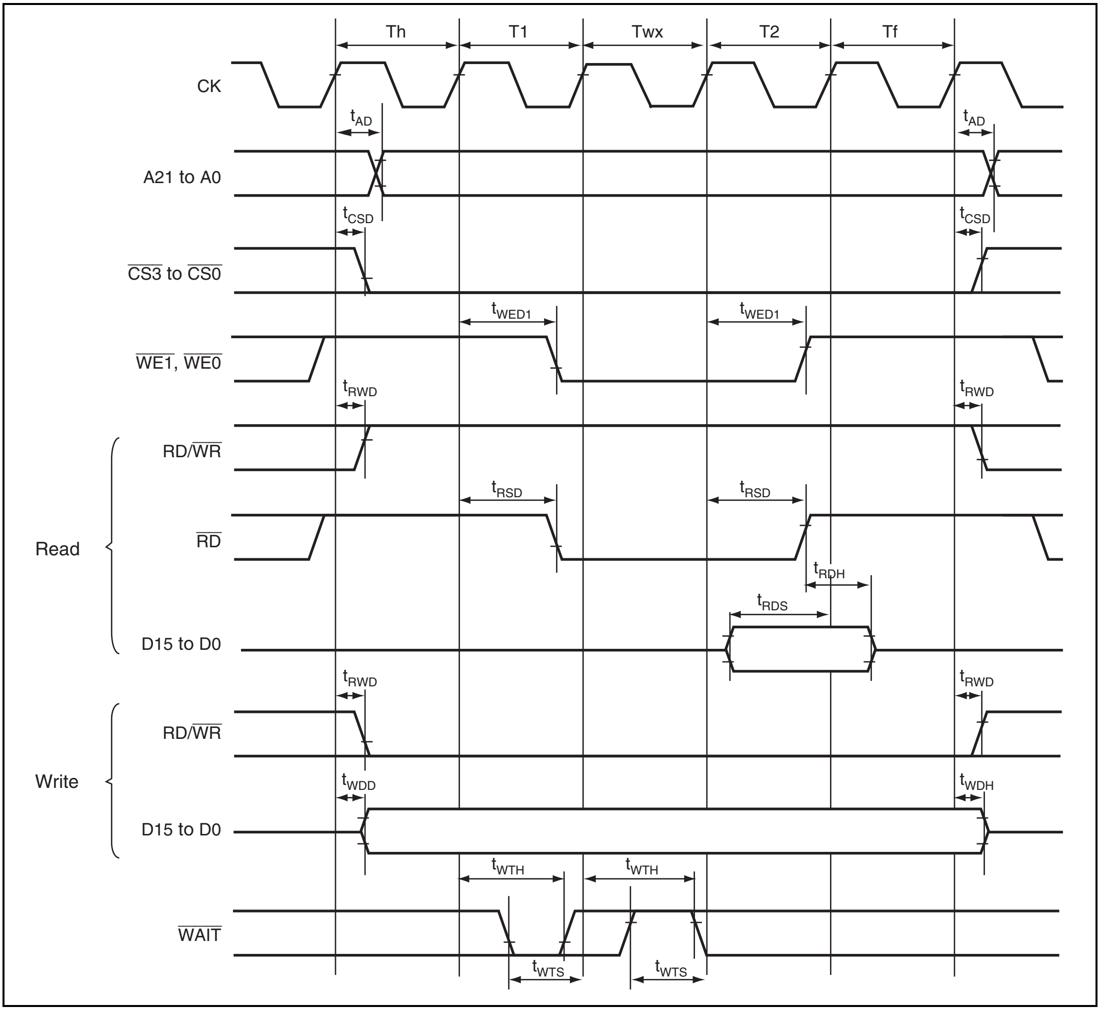


Figure 34.12 Bus Timing of SRAM with Byte Selection (SW = 1 Cycle, HW = 1 Cycle, One Asynchronous External Wait Cycle, BAS = 0 (Write Cycle UB/LB Control))

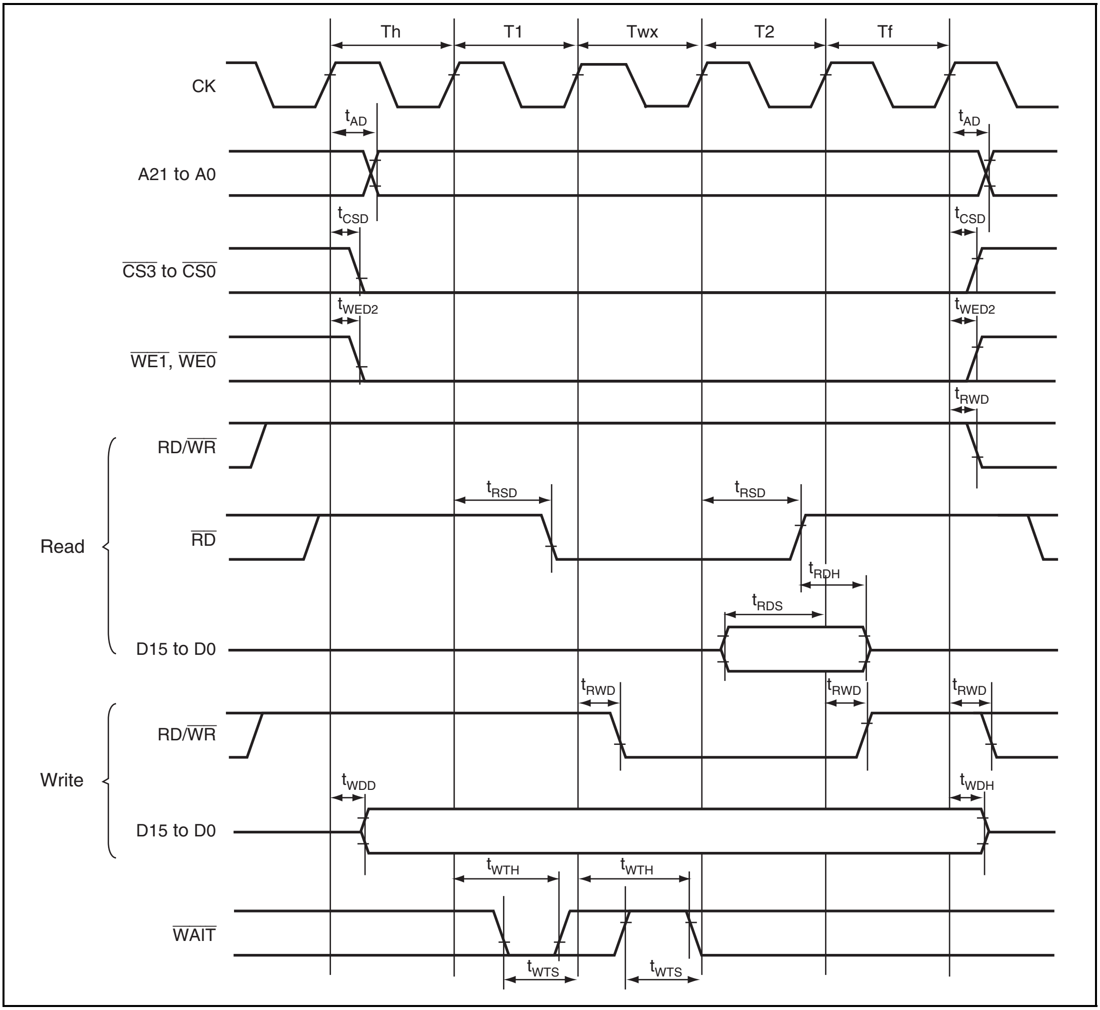


Figure 34.13 Bus Timing of SRAM with Byte Selection (SW = 1 Cycle, HW = 1 Cycle, One Asynchronous External Wait Cycle, BAS = 1 (Write Cycle WE Control))

34.3.6 Advanced Timer Unit Timing and Advanced Pulse Controller Timing

Table 34.18 shows the advanced timer unit timing and advanced pulse controller timing.

Table 34.18 Advanced Timer Unit Timing

Conditions: V_{CC} = PLLV_{CC} = 3.3 V ±0.3 V, PV_{CC}1 = 5.0 V ±0.5 V/3.3 V ±0.3 V,
PV_{CC}2 = 5.0 V ±0.5 V,
AV_{CC} = 5.0 V ±0.5 V, AVREFH_A = AVREFH_B = 4.5 V to AV_{CC},
V_{SS} = PLLV_{SS} = AV_{SS} = AVREFL_A = AVREFL_B = 0 V
When PV_{CC}1 = 3.3 V ±0.3 V, V_{CC} = PV_{CC}1. T_a = –40°C to 125°C

Item		Symbol	Min.	Typ.*[1]	Max.	Unit	Figures
Output compare output delay time		t_{TOCD}	—	—	100	ns	Figure 34.14
Output compare output rise/fall time*[2]	Slow	t_{SR}, t_{SF}	—	50	100	ns	Figure 34.29
	Fast	t_{FR}, t_{FF}	—	6	12		
Input capture input setup time		t_{TICS}	24	—	—	ns	Figure 34.14
Timer clock input setup time		t_{TCKS}	24	—	—	ns	Figure 34.15
Timer clock pulse width (single edge specified)		t_{TCKWH} t_{TCKWL}	1.5	—	—	t_{cyc}	
Timer clock pulse width (both edges specified)		t_{TCKWH} t_{TCKWL}	2.5	—	—	t_{cyc}	

Notes: 1. The Typ condition is V_{CC} = 3.3V, PV_{CC}1 = PV_{CC}2 = 5.0V, T_a = 25°C.
2. Refer to table 34.2 for the pins that can be switched to the faster slew rate.

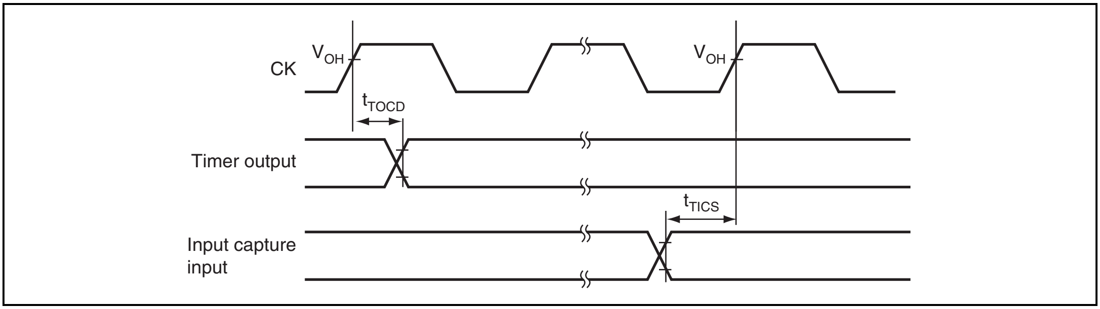


Figure 34.14 ATU Input/Output Timing

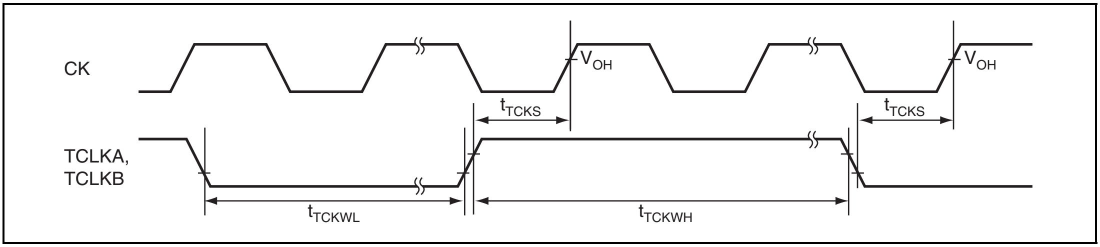


Figure 34.15 ATU Clock Input Timing

34.3.7 I/O Port Timing

Table 34.19 shows the I/O port timing.

Table 34.19 I/O Port Timing

Conditions: V_{CC} = PLLV_{CC} = 3.3 V ±0.3 V, PV_{CC}1 = 5.0 V ±0.5 V/3.3 V ±0.3 V,
PV_{CC}2 = 5.0 V ±0.5 V,
AV_{CC} = 5.0 V ±0.5 V, AVREFH_A = AVREFH_B = 4.5 V to AV_{CC},
V_{SS} = PLLV_{SS} = AV_{SS} = AVREFL_A = AVREFL_B = 0 V
When PV_{CC}1 = 3.3 V ±0.3 V, V_{CC} = PV_{CC}1.
T_a = –40°C to 125°C

Item		Symbol	Min.	Typ.*1	Max.	Unit	Figures
Port output data delay time		t_{PWD}	—	—	100	ns	Figure 34.16
Port output data rise/fall time*2	Slow	t_{SR}, t_{SF}	—	50	100	ns	Figure 34.29
	Fast	t_{FR}, t_{FF}	—	6	12		
Port input setup time		t_{PRS}	24	—	—	ns	Figure 34.16
Port input hold time (level-input except for port G)		t_{PRH}	24	—	—	ns	
Port input hold time (port G edge-input)		t_{PRH}	t_{cyc} + 24	—	—	ns	

Notes: 1. The Typ condition is V_{CC} = 3.3V, PV_{CC}1 = PV_{CC}2 = 5.0V, T_a = 25°C.
2. Refer to table 34.2 for the pins that can be switched to the faster slew rate.

[Operating precautions]

The guaranteed operating range of power supply PV_{CC}1 in MCU single-chip mode is only PV_{CC}1 = 5.0 V ±0.5 V. Do not use a voltage outside this range.

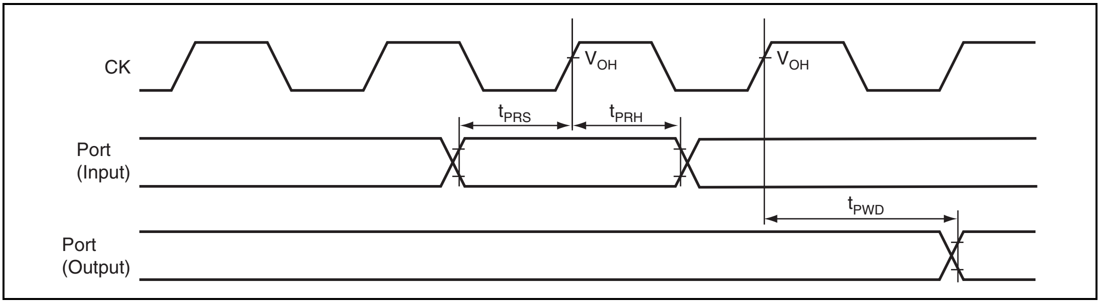


Figure 34.16 I/O Port Input/Output Timing

34.3.8 Watchdog Timer Timing

Table 34.20 shows the watchdog timer timing.

Table 34.20 Watchdog Timer Timing

Conditions: V_{CC} = $PLLV_{CC}$ = 3.3 V ±0.3 V, $PV_{CC}1$ = 5.0 V ±0.5 V/3.3 V ±0.3 V,
$PV_{CC}2$ = 5.0 V ±0.5 V,
AV_{CC} = 5.0 V ±0.5 V, AVREFH_A = AVREFH_B = 4.5 V to AV_{CC},
V_{SS} = $PLLV_{SS}$ = AV_{SS} = AVREFL_A = AVREFL_B = 0 V
When $PV_{CC}1$ = 3.3 V ±0.3 V, V_{CC} = $PV_{CC}1$.
T_a = –40°C to 125°C

Item	Symbol	Min.	Max.	Unit	Figures
$\overline{\text{WDTOVF}}$ delay time	t_{WOVD}	—	100	ns	Figure 34.17

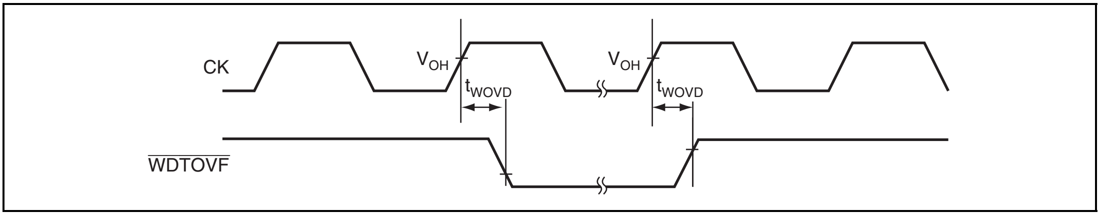


Figure 34.17 Watchdog Timer Timing

34.3.9 Serial Communications Interface Timing

Table 34.21 shows the serial communications interface timing.

Table 34.21 Serial Communications Interface Timing

Conditions: V_{CC} = PLLV_{CC} = 3.3 V ±0.3 V, $PV_{CC}1$ = 5.0 V ±0.5 V/3.3 V ±0.3 V,
$PV_{CC}2$ = 5.0 V ±0.5 V,
AV_{CC} = 5.0 V ±0.5 V, AVREFH_A = AVREFH_B = 4.5 V to AV_{CC},
V_{SS} = PLLV_{SS} = AV_{SS} = AVREFL_A = AVREFL_B = 0 V
When $PV_{CC}1$ = 3.3 V ±0.3 V, V_{CC} = $PV_{CC}1$.
T_a = –40°C to 125°C

Item		Symbol	Min.	Typ.*	Max.	Unit	Figures
Clock cycle (clock sync: input)	MD_CLKP = 0	t_{scyc}	12	—	—	t_{cyc}	Figure 34.18
Clock cycle (clock sync: output)		t_{scyc}	8	—	—	t_{cyc}	
Clock cycle (clock sync: input)	MD_CLKP = 1	t_{scyc}	16	—	—	t_{cyc}	
Clock cycle (clock sync: output)		t_{scyc}	8	—	—	t_{cyc}	
Clock pulse width		t_{sckw}	0.4	—	0.6	t_{scyc}	
Input clock rise time	MD_CLKP = 0	t_{sckr}	—	—	0.8	t_{cyc}	
Input clock fall time		t_{sckf}	—	—	0.8	t_{cyc}	
Input clock rise time	MD_CLKP = 1	t_{sckr}	—	—	1.6	t_{cyc}	
Input clock fall time		t_{sckf}	—	—	1.6	t_{cyc}	
Synchronous mode transmit data delay time (SCK is input)	MD_CLKP = 0	t_{TxD}	—	—	3 × t_{cyc} + 100	ns	Figure 34.19
	MD_CLKP = 1		—	—	3 × t_{cyc} + 75		
Synchronous mode receive data setup time (SCK is input)		t_{RxS}	–1 × t_{cyc} + 30	—	—	ns	
Synchronous mode receive data hold time (SCK is input)		t_{RxH}	2 × t_{cyc} + 30	—	—	ns	
Synchronous mode transmit data delay time (SCK is output)		t_{TxD}	—	—	70	ns	

Item			Symbol	Min.	Typ.*	Max.	Unit	Figures
Synchronous mode receive data setup time (SCK is output)	Transmission and reception, or transmission only	Slow	t_{RxS}	120	—	—	ns	Figure 34.19
		Fast		30	—	—		
	Reception only	Slow		120 + 3 × t_{cyc}	—	—		
		Fast		30 + 3 × t_{cyc}	—	—		
Synchronous mode receive data hold time (SCK is output)			t_{RxH}	1 × t_{cyc} + 30	—	—	ns	
Asynchronous mode transmit data delay time			t_{TxD}	—	—	100	ns	
Asynchronous mode receive data setup time			t_{RxS}	100	—	—	ns	
Asynchronous mode receive data hold time			t_{RxH}	100	—	—	ns	
SCK clock output rise/fall time		Slow	t_{SR}, t_{SF}	—	50	100	ns	Figure 34.29
		Fast	t_{FR}, t_{FF}	—	6	12		
TxD Transmit data output rise/fall time		Slow	t_{SR}, t_{SF}	—	50	100	ns	
		Fast	t_{FR}, t_{FF}	—	6	12		

Note: * The Typ condition is V_{CC} = 3.3V, $PV_{CC}1$ = $PV_{CC}2$ = 5.0V, T_a = 25°C.

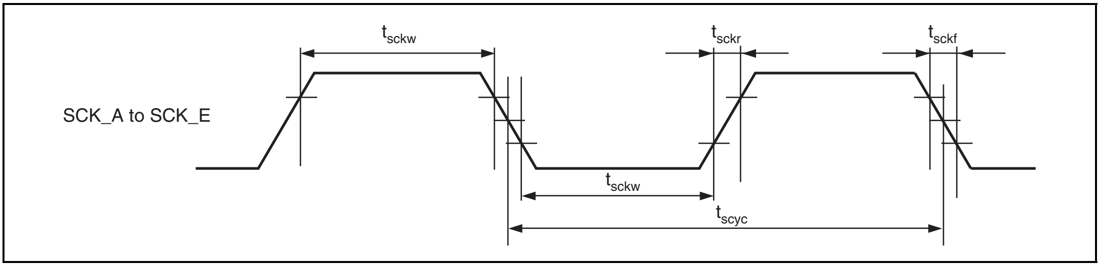


Figure 34.18 SCK Input Timing

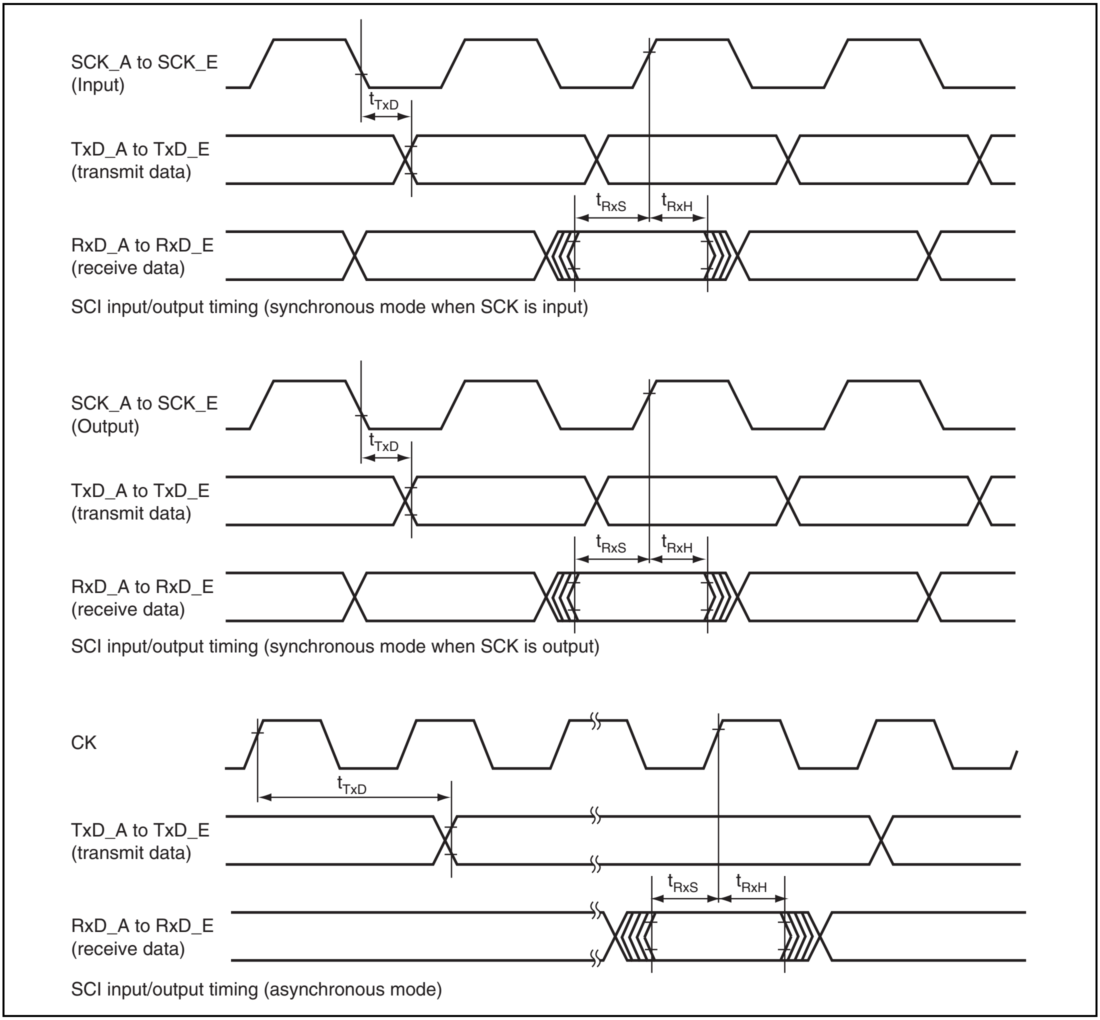


Figure 34.19 SCI Input/Output Timing

34.3.10 CAN Timing

Table 34.22 shows the CAN timing.

Table 34.22 CAN Timing

Conditions: V_{CC} = $PLLV_{CC}$ = 3.3 V ±0.3 V, $PV_{CC}1$ = 5.0 V ±0.5 V/3.3 V ±0.3 V,
$PV_{CC}2$ = 5.0 V ±0.5 V,
AV_{CC} = 5.0 V ±0.5 V, AVREFH_A = AVREFH_B = 4.5 V to AV_{CC},
V_{SS} = $PLLV_{SS}$ = AV_{SS} = AVREFL_A = AVREFL_B = 0 V
When $PV_{CC}1$ = 3.3 V ±0.3 V, V_{CC} = $PV_{CC}1$.
T_a = –40°C to 125°C

Item		Symbol	Min.	Typ.*[1]	Max.	Unit	Figures
CAN Transmit data delay time		t_{CTxD}	—	—	100	ns	Figure 34.20
CTx Transmit data output rise/fall time	Slow	t_{SR}, t_{SF}	—	50	100	ns	Figure 34.29
	Fast	t_{FR}, t_{FF}	—	6	12		
CAN Receive data setup time		t_{CRxS}	100	—	—	ns	Figure 34.20
CAN Receive data hold time		t_{CRxH}	100	—	—	ns	
CAN bus jitter*[3]	Period 1µs	f_{CANJIT}	—	0.13*[2]	—	%	1Mbps/1-bit
	Period 2µs		—	0.08*[2]	0.1	%	500kbps/1-bit
	Period 10µs		—	0.02*[2]	—	%	1Mbps/10-bit
	Period 20µs		—	0.01*[2]	0.1	%	500kbps/10-bit

Notes: 1. The Typ condition is V_{CC} = 3.3V, $PV_{CC}1$ = $PV_{CC}2$ = 5.0V, T_a = 25°C.
2. The typ value of CAN bus jitter is measured value of the representative sample without crystal deviation.
It is not a guarantee value to which all samples are tested.
3. The crystal deviation is not included.

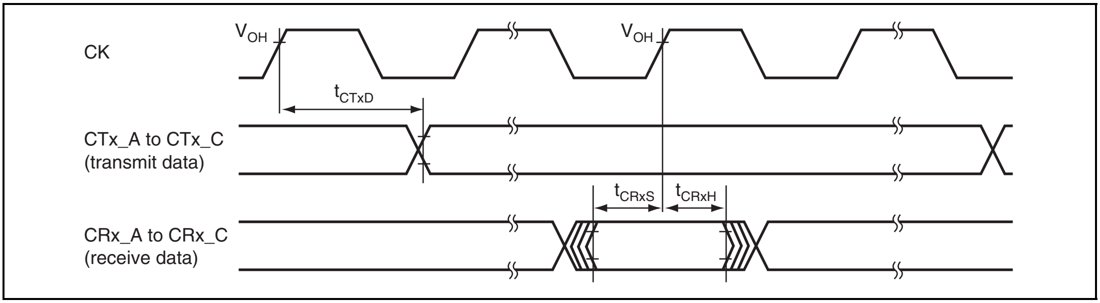


Figure 34.20 CAN Input/Output Timing

34.3.11 SPI Timing

Tables 34.23 (1) and 34.23 (2) show the SPI timing.

Table 34.23 (1) SPI Timing (when Output Slew Rates are Fast)

Conditions: V_{CC} = PLLV_{CC} = 3.3 V ±0.3 V, PV_{CC}1 = 5.0 V ±0.5 V/3.3 V ±0.3 V,
PV_{CC}2 = 5.0 V ±0.5 V,
AV_{CC} = 5.0 V ±0.5 V, AVREFH_A = AVREFH_B = 4.5 V to AV_{CC},
V_{SS} = PLLV_{SS} = AV_{SS} = AVREFL_A = AVREFL_B = 0 V
When PV_{CC}1 = 3.3 V ±0.3 V, V_{CC} = PV_{CC}1.
T_a = –40°C to 125°C

Item			Symbol	Min.	Typ.*	Max.	Unit	Figures
RSPCK clock cycle	MD_CLKP = 0	Master	t_{SPcyc}	2	—	4096	t_{cyc}	Figure 34.21
		Slave		8	—	4096		
	MD_CLKP = 1	Master		4	—	4096	t_{cyc}	
		Slave		16	—	4096		
RSPCK clock high-level pulse width		Master	t_{SPCKWH}	$(t_{SPcyc} - t_{SPCKR} - t_{SPCKF})/2 - 5$	—	—	ns	
		Slave		$(t_{SPcyc} - t_{SPCKR} - t_{SPCKF})/2$	—	—	ns	
RSPCK clock low-level pulse width		Master	t_{SPCKWL}	$(t_{SPcyc} - t_{SPCKR} - t_{SPCKF})/2 - 5$	—	—	ns	
		Slave		$(t_{SPcyc} - t_{SPCKR} - t_{SPCKF})/2$	—	—	ns	
RSPCK clock rise/fall time		Output	t_{SPCKR}, t_{SPCKF}	—	3	5	ns	
		Input		—	—	1	µs	

Item		Symbol	Min.	Typ.*	Max.	Unit	Figures
Data input setup time	Master	t_{SU}	18	—	—	ns	Figures 34.22 to 34.25
	Slave		$20 - 2 \times t_{cyc}$	—	—	ns	
Data input hold time	Master	t_{H}	0	—	—	ns	
	Slave		$20 + 2 \times t_{cyc}$	—	—		
SSL setup time	Master	t_{LEAD}	1	—	8	t_{SPcyc}	
	Slave		4	—	—	t_{cyc}	
SSL hold time	Master	t_{LAG}	1	—	8	t_{SPcyc}	
	Slave		4	—	—	t_{cyc}	
Data output delay time	Master	t_{OD}	—	—	15	ns	
	Slave		—	—	$25 + 3 \times t_{cyc}$		
Data output hold time	Master	t_{OH}	0	—	—	ns	
	Slave		0	—	—		
Continuous transmission delay time	Master	t_{TD}	$t_{SPcyc} + 2 \times t_{cyc}$	—	$8 \times t_{SPcyc} + 2 \times t_{cyc}$	ns	
	Slave		$4 \times t_{cyc}$	—	—		
MOSI, MISO rise/fall time	Output	$t_{DR,}$	—	3	5	ns	
	Input	t_{DF}	—	—	1	μs	
SSL rise/fall time	Output	$t_{SSLR,}$	—	3	5	ns	
	Input	t_{SSLF}	—	—	1	μs	
Slave access time		t_{SA}	—	—	4	t_{cyc}	Figures 34.24 and 34.25
Slave out release time		t_{REL}	—	—	3	t_{cyc}	

Note: * The Typ condition is $V_{CC} = 3.3V$, $PV_{CC}1 = PV_{CC}2 = 5.0V$, $T_a = 25°C$.
Timing on condition that the slew rates of RSPCK, MISO/MOSI, and SSL are all fast.

Table 34.23 (2) SPI Timing (when Output Slew Rates are Slow)

Conditions: V_{CC} = PLLV_{CC} = 3.3 V ±0.3 V, PV_{CC}1 = 5.0 V ±0.5 V/3.3 V ±0.3 V,
PV_{CC}2 = 5.0 V ±0.5 V,
AV_{CC} = 5.0 V ±0.5 V, AVREFH_A = AVREFH_B = 4.5 V to AV_{CC},
V_{SS} = PLLV_{SS} = AV_{SS} = AVREFL_A = AVREFL_B = 0 V
When PV_{CC}1 = 3.3 V ±0.3 V, V_{CC} = PV_{CC}1.
T_a = –40°C to 125°C

Item			Symbol	Min.	Typ.*	Max.	Unit	Figures
RSPCK clock cycle	MD_CLKP = 0	Master	t_{SPcyc}	8	—	4096	t_{cyc}	Figure 34.21
		Slave		8	—	4096		
	MD_CLKP = 1	Master		16	—	4096	t_{cyc}	
		Slave		16	—	4096		
RSPCK clock high-level pulse width		Master	t_{SPCKWH}	$(t_{SPcyc} - t_{SPCKR} - t_{SPCKF})/2 - 5$	—	—	ns	
		Slave		$(t_{SPcyc} - t_{SPCKR} - t_{SPCKF})/2$	—	—		
RSPCK clock low-level pulse width		Master	t_{SPCKWL}	$(t_{SPcyc} - t_{SPCKR} - t_{SPCKF})/2 - 5$	—	—	ns	
		Slave		$(t_{SPcyc} - t_{SPCKR} - t_{SPCKF})/2$	—	—		
RSPCK clock rise/fall time		Output	t_{SPCKR}, t_{SPCKF}	—	20	40	ns	
		Input		—	—	1	µs	

Item		Symbol	Min.	Typ.*	Max.	Unit	Figures
Data input setup time	Master	t_{SU}	100	—	—	ns	Figures 34.22 to 34.25
	Slave		20 – 2 × t_{cyc}	—	—	ns	
Data input hold time	Master	t_H	0	—	—	ns	
	Slave		20 + 2 × t_{cyc}	—	—		
SSL setup time	Master	t_{LEAD}	1	—	8	t_{SPcyc}	
	Slave		4	—	—	t_{cyc}	
SSL hold time	Master	t_{LAG}	1	—	8	t_{SPcyc}	
	Slave		4	—	—	t_{cyc}	
Data output delay time	Master	t_{OD}	—	—	20	ns	
	Slave		—	—	100 + 3 × t_{cyc}		
Data output hold time	Master	t_{OH}	–10	—	—	ns	
	Slave		0	—	—		
Continuous transmission delay time	Master	t_{TD}	t_{SPcyc} + 2 × t_{cyc}	—	8 × t_{SPcyc} + 2 × t_{cyc}	ns	
	Slave		4 × t_{cyc}	—	—		
MOSI, MISO rise/fall time	Output	$t_{DR,}$	—	20	40	ns	
	Input	t_{DF}	—	—	1	μs	
SSL rise/fall time	Output	$t_{SSLR,}$	—	20	40	ns	
	Input	t_{SSLF}	—	—	1	μs	
Slave access time		t_{SA}	—	—	4	t_{cyc}	Figures 34.24 and 34.25
Slave out release time		t_{REL}	—	—	3	t_{cyc}	

Note: * The Typ condition is V_{CC} = 3.3V, $PV_{CC}1$ = $PV_{CC}2$ = 5.0V, T_a = 25°C.
Timing on condition that the slew rates of RSPCK, MISO/MOSI, and SSL are all slow.

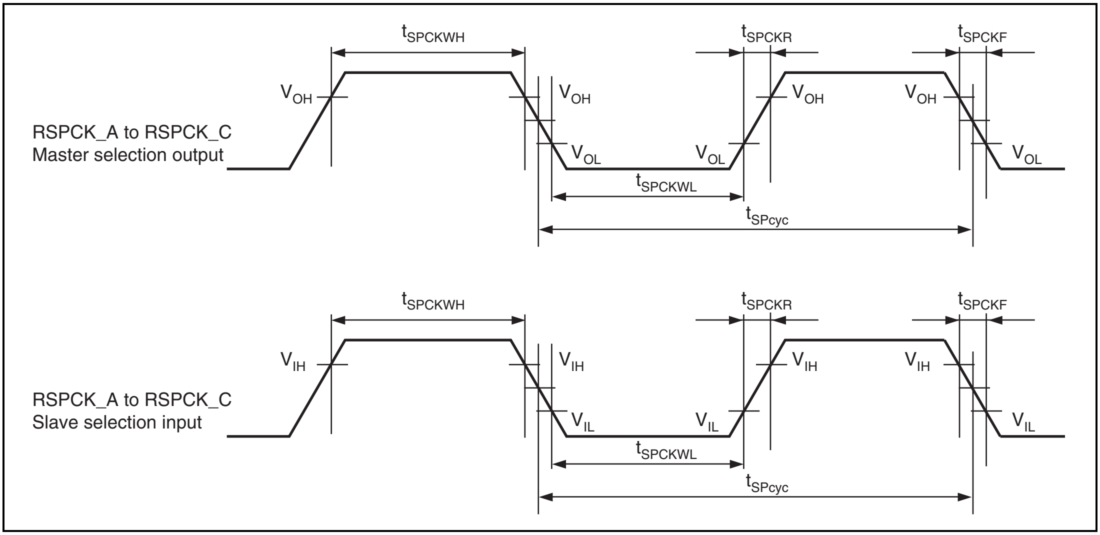


Figure 34.21 SPI Clock Timing

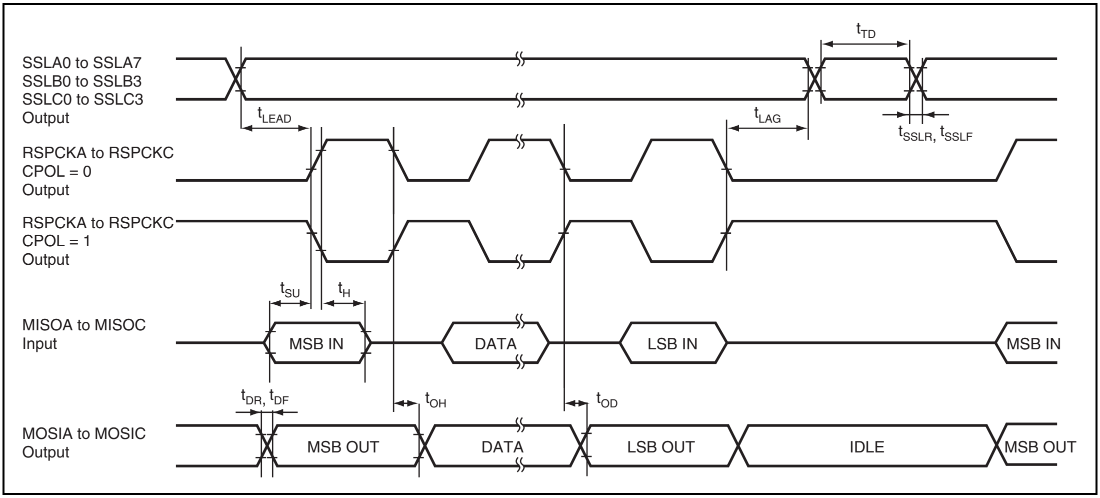


Figure 34.22 SPI Timing (Master, CPHA = 0)

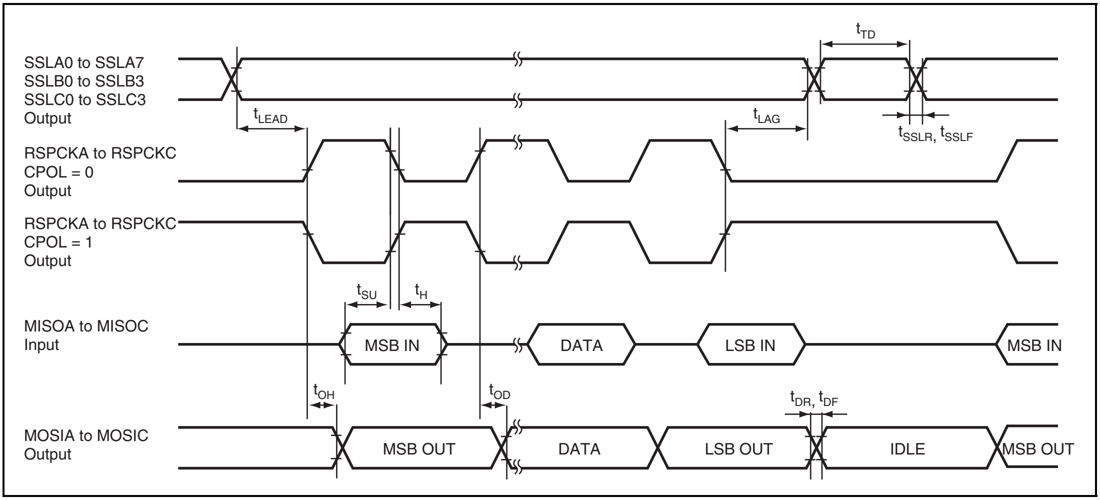


Figure 34.23 SPI Timing (Master, CPHA = 1)

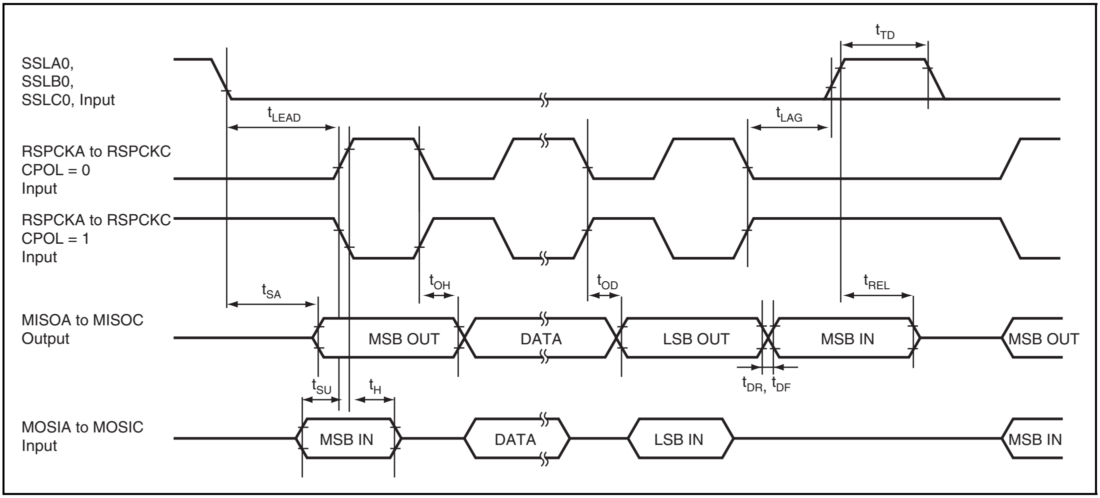


Figure 34.24 SPI Timing (Slave, CPHA = 0)

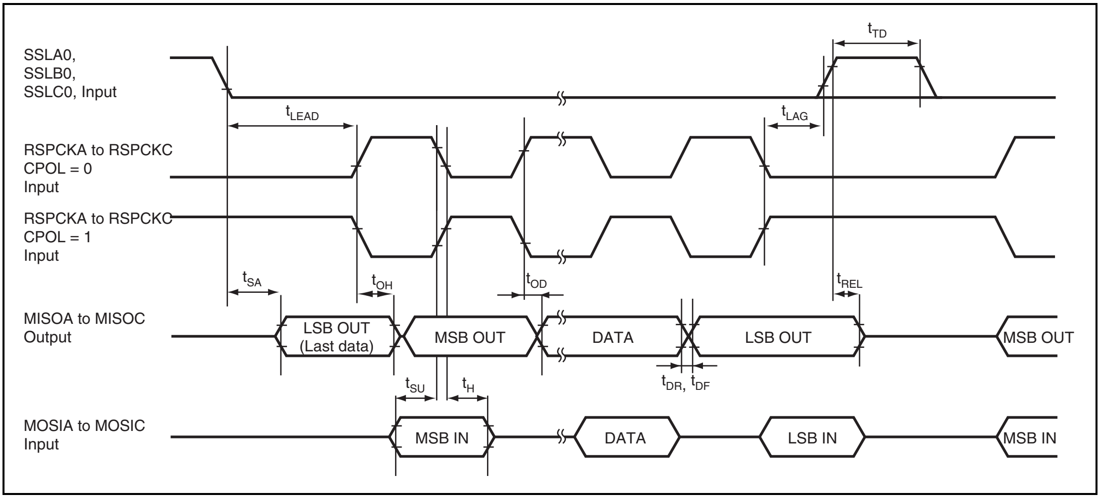


Figure 34.25 SPI Timing (Slave, CPHA = 1)

34.3.12 A/D Converter Timing

Table 34.24 shows the A/D converter timing.

Table 34.24 A/D Converter Timing

Conditions: V_{CC} = $PLLV_{CC}$ = 3.3 V ±0.3 V, $PV_{CC}1$ = 5.0 V ±0.5 V/3.3 V ±0.3 V,
$PV_{CC}2$ = 5.0 V ±0.5 V,
AV_{CC} = 5.0 V ±0.5 V, AVREFH_A = AVREFH_B = 4.5 V to AV_{CC},
V_{SS} = $PLLV_{SS}$ = AV_{SS} = AVREFL_A = AVREFL_B = 0 V
When $PV_{CC}1$ = 3.3 V ±0.3 V, V_{CC} = $PV_{CC}1$.
T_a = –40°C to 125°C

Item		Symbol	Min.	Typ.*	Max.	Unit	Figure
External trigger input start delay time		t_{TRGS}	50	—	—	ns	Figure 34.26
ADEND output delay time		t_{ADENDD}	—	—	100	ns	Figure 34.27
ADEND output rise/fall time	Slow	t_{SR}, t_{SF}	—	—	100	ns	Figure 34.29
	Fast	t_{FR}, t_{FF}	—	—	12		

Note: * The Typ condition is V_{CC} = 3.3V, $PV_{CC}1$ = $PV_{CC}2$ = 5.0V, T_a = 25°C.

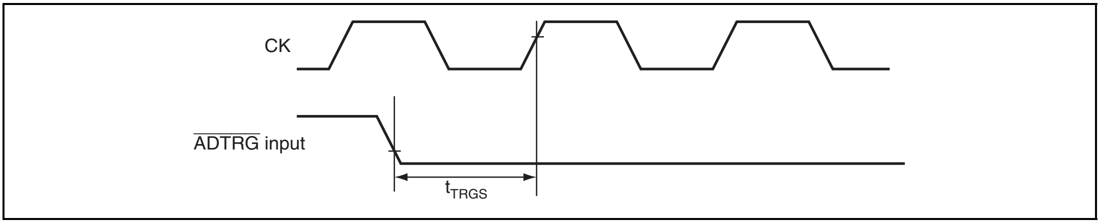


Figure 34.26 External Trigger Input Timing

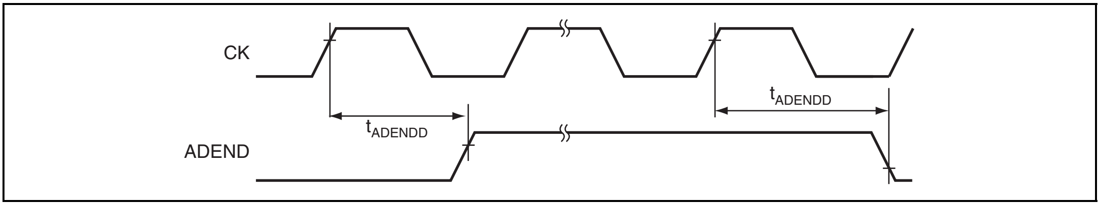


Figure 34.27 Analog Conversion Timing

34.3.13 UBC Trigger Timing

Table 34.25 shows the UBC trigger timing.

Table 34.25 UBC Trigger Timing

Conditions: V_{CC} = $PLLV_{CC}$ = 3.3 V ±0.3 V, $PV_{CC}1$ = 5.0 V ±0.5 V/3.3 V ±0.3 V,
$PV_{CC}2$ = 5.0 V ±0.5 V,
AV_{CC} = 5.0 V ±0.5 V, AVREFH_A = AVREFH_B = 4.5 V to AV_{CC},
V_{SS} = $PLLV_{SS}$ = AV_{SS} = AVREFL_A = AVREFL_B = 0 V
When $PV_{CC}1$ = 3.3 V ±0.3 V, V_{CC} = $PV_{CC}1$.
T_a = –40°C to 125°C

Item		Symbol	Min.	Typ.*	Max.	Unit	Figures
$\overline{\text{UBCTRG}}$ delay time		t_{UBCTGD}	—	—	100	ns	Figure 34.28
$\overline{\text{UBCTRG}}$ output rise/fall time	Slow	t_{SR}, t_{SF}	—	50	100	ns	Figure 34.29
	Fast	t_{FR}, t_{FF}	—	6	12		

Note: * The Typ condition is V_{CC} = 3.3V, $PV_{CC}1$ = $PV_{CC}2$ = 5.0V, T_a = 25°C.

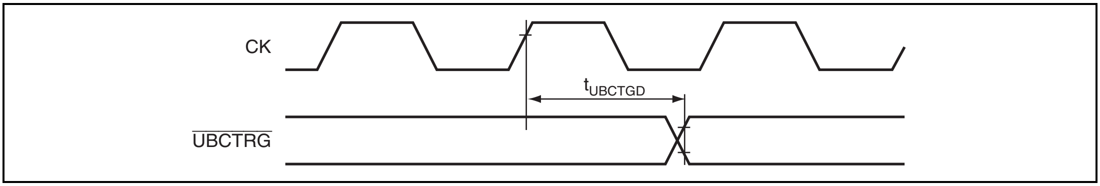


Figure 34.28 UBC Trigger Timing

34.3.14 Output Slew Rate

Table 34.26 shows the output slew rate timing.

Table 34.26 Output Slew Rate Timing

Conditions: V_{CC} = PLLV_{CC} = 3.3 V ±0.3 V, PV_{CC}1 = 5.0 V ±0.5 V/3.3 V ±0.3 V,
PV_{CC}2 = 5.0 V ±0.5 V,
AV_{CC} = 5.0 V ±0.5 V, AVREFH_A = AVREFH_B = 4.5 V to AV_{CC},
V_{SS} = PLLV_{SS} = AV_{SS} = AVREFL_A = AVREFL_B = 0 V
When PV_{CC}1 = 3.3 V ±0.3 V, V_{CC} = PV_{CC}1.
T_a = –40°C to 125°C

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit	Figures
Output rise/fall time Slow slew rate	t_{SR}, t_{SF}	Load 25[pF]	—	25	50	ns	Figure 34.29
		Load 50[pF]	—	50	100		
		Load 75[pF]	—	75	150		
		Load 100[pF]	—	100	200		
Output rise/fall time Fast slew rate	t_{FR}, t_{FF}	Load 25[pF]	—	4	6	ns	
		Load 50[pF]	—	6	12		
		Load 75[pF]	—	8	18		
		Load 100[pF]	—	10	24		

Notes: 1. The Typ condition is V_{CC} = 3.3V, PV_{CC}1 = PV_{CC}2 = 5.0V, T_a = 25°C.
2. Refer to table 34.2 for the pins that can be switched to the faster slew rate.

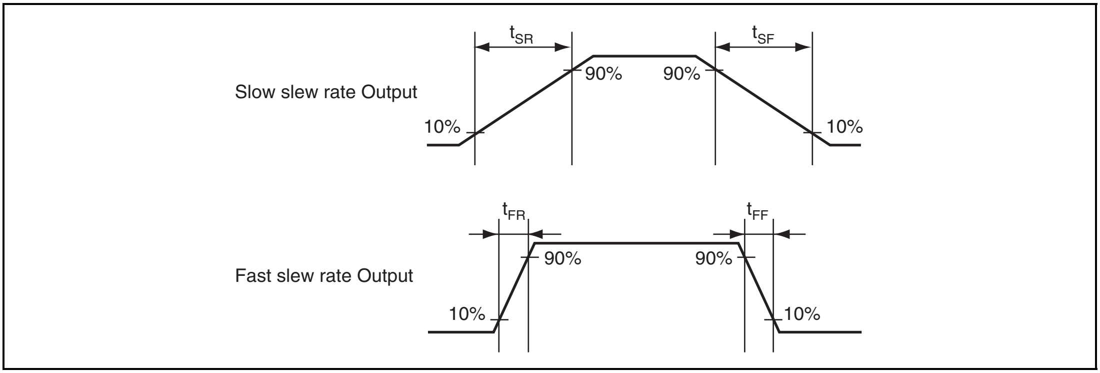


Figure 34.29 Output Slew Rate Timing

34.3.15 FlexRay Timing

Table 34.27 shows the FlexRay timing.

Table 34.27 FlexRay Timing

Conditions: V_{CC} = $PLLV_{CC}$ = 3.3 V ±0.3 V, $PV_{CC}1$ = 5.0 V ±0.5 V/3.3 V ±0.3 V,
$PV_{CC}2$ = 5.0 V ±0.5 V,
AV_{CC} = 5.0 V ±0.5 V, AVREFH_A = AVREFH_B = 4.5 V to AV_{CC},
V_{SS} = $PLLV_{SS}$ = AV_{SS} = AVREFL_A = AVREFL_B = 0 V
When $PV_{CC}1$ = 3.3 V ±0.3 V, V_{CC} = $PV_{CC}1$.
T_a = –40°C to 125°C

Item		Symbol	Min.	Typ. *[1]	Max.	Unit	Figures
FRTxD transmit data rise/fall time *[2]	C_L = 15 [pF]	t_{TxDR}, t_{TxDF}	—	—	2.5	ns	Figure 34.30

Notes: 1. The Typ condition is V_{CC} = 3.3V, $PV_{CC}1$ =5.0/3.3 V, $PV_{CC}2$ = 5.0V, T_a = 25°C.
2. The rising/falling characteristics apply to all functions when the pins associated with the FlexRay have been set to select higher driving ability. For details on functions of the multiplxed pins, refer to section 23, Pin Function Controller (PFC).

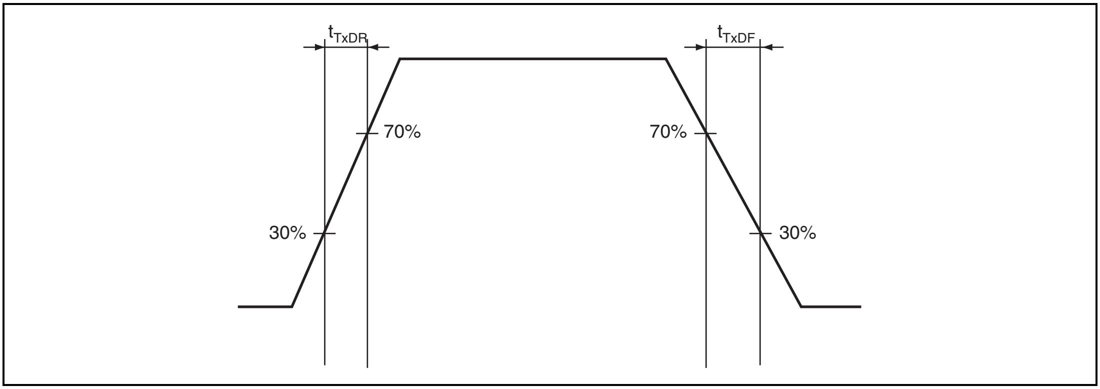


Figure 34.30 FlexRay Transmit Slew Rate Timing

34.3.16 JTAG Interface Timing

Table 34.28 shows the JTAG interface timing.

Table 34.28 JTAG Interface Timing

Conditions: V_{CC} = $PLLV_{CC}$ = 3.3 V ±0.3 V, $PV_{CC}1$ = 5.0 V ±0.5 V/3.3 V ±0.3 V,
$PV_{CC}2$ = 5.0 V ±0.5 V,
AV_{CC} = 5.0 V ±0.5 V, AVREFH_A = AVREFH_B = 4.5 V to AV_{CC},
V_{SS} = $PLLV_{SS}$ = AV_{SS} = AVREFL_A = AVREFL_B = 0 V
When $PV_{CC}1$ = 3.3 V ±0.3 V, V_{CC} = $PV_{CC}1$.
T_a = –40°C to 125°C

Item		Symbol	Min.	Max.	Unit	Figures
TCK clock cycle	MD_CLKP = 0	t_{TCKcyc}	10	—	t_{cyc}	Figure 34.31
	MD_CLKP = 1		20	—		
TCK clock high-level width		t_{TCKH}	0.4	0.6	t_{TCKcyc}	
TCK clock low-level width		t_{TCKL}	0.4	0.6	t_{TCKcyc}	
$\overline{\text{TRST}}$ pulse width		t_{TRSW}	20	—	t_{TCKcyc}	Figure 34.32
TMS setup time		t_{TMSS}	30	—	ns	Figure 34.33
TMS hold time		t_{TMSH}	30	—	ns	
TDI setup time		t_{TDIS}	30	—	ns	
TDI hold time		t_{TDIH}	30	—	ns	
TDO delay time		t_{TDOD}	—	45	ns	

[Operating precautions]

The JTAG Interface pins constitute a circuit requiring the voltage of V_{CC} = 3.3 V ±0.3 V. Comply with the input and output voltages specified in the DC characteristics, for operation.

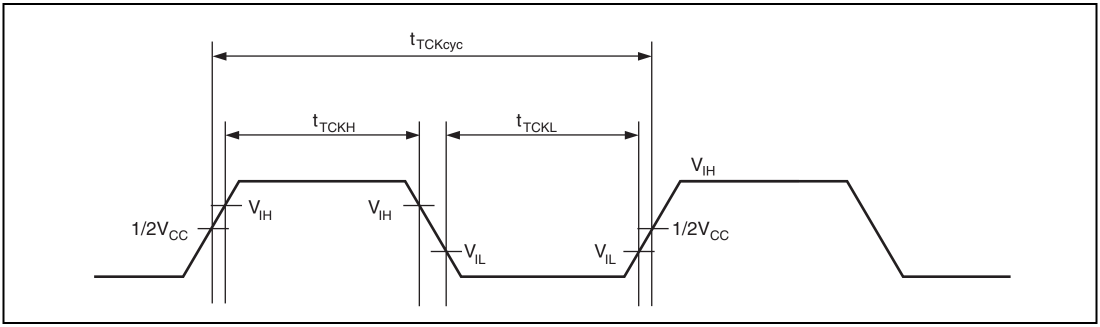


Figure 34.31 JTAG Interface Clock Timing

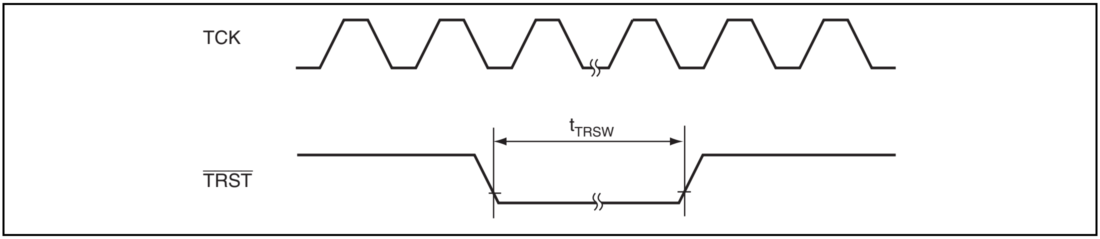


Figure 34.32 JTAG Interface $\overline{\text{TRST}}$ Timing

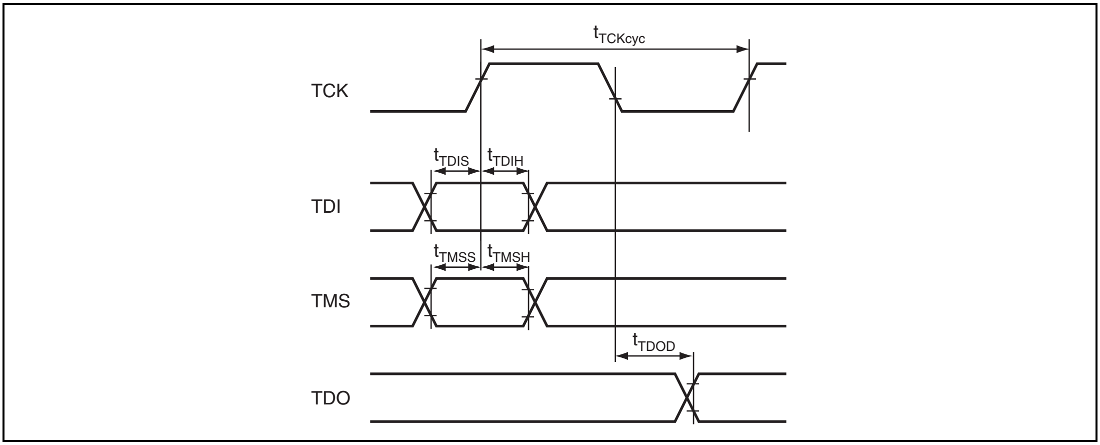


Figure 34.33 JTAG Interface Input/Output

34.3.17 AUD Timing

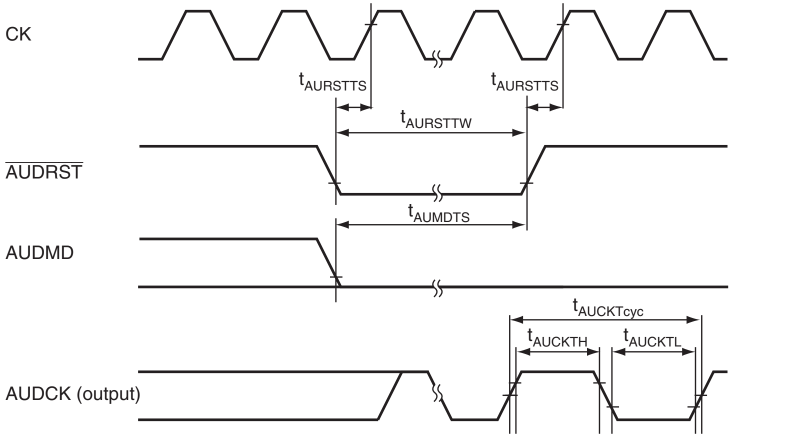
Table 34.29 shows the AUD timing.

Table 34.29 AUD Timing

Conditions: V_{CC} = PLLV_{CC} = 3.3 V ±0.3 V, PV_{CC}1 = 5.0 V ±0.5 V/3.3 V ±0.3 V,
PV_{CC}2 = 5.0 V ±0.5 V,
AV_{CC} = 5.0 V ±0.5 V, AVREFH_A = AVREFH_B = 4.5 V to AV_{CC},
V_{SS} = PLLV_{SS} = AV_{SS} = AVREFL_A = AVREFL_B = 0 V
When PV_{CC}1 = 3.3 V ±0.3 V, V_{CC} = PV_{CC}1.
T_a = –40°C to 125°C

Item		Symbol	Min.	Max.	Unit	Figures
$\overline{\text{AUDRST}}$ setup time (trace mode)		$t_{AURSTTS}$	30	—	ns	Figure 34.34
$\overline{\text{AUDRST}}$ pulse width (trace mode)	MD_CLKP = 0	$t_{AURSTTW}$	10	—	t_{cyc}	
	MD_CLKP = 1		20	—		
AUDMD setup time (trace mode)	MD_CLKP = 0	t_{AUMDTS}	10	—	t_{cyc}	
	MD_CLKP = 1		20	—		
AUDCK cycle time (trace mode)	MD_CLKP = 0	$t_{AUCKTcyc}$	1	2.5	t_{cyc}	
	MD_CLKP = 1		2	5		
AUDCK high-level width (trace mode)		t_{AUCKTH}	0.4	0.6	$t_{AUCKTcyc}$	
AUDCK low-level width (trace mode)		t_{AUCKTL}	0.4	0.6	$t_{AUCKTcyc}$	
Trace data output delay time		t_{AUDTTD}	—	20	ns	Figure 34.35
$\overline{\text{AUDSYNC}}$ output delay time		t_{AUSYTD}	—	20	ns	
AUDCK cycle time (monitor mode)	MD_CLKP = 0	$t_{AUCKMcyc}$	1	—	t_{cyc}	Figure 34.36
	MD_CLKP = 1		2	—		
AUDCK high-level width (monitor mode)		t_{AUCKMH}	0.4	—	$t_{AUCKMcyc}$	
AUDCK low-level width (monitor mode)		t_{AUCKML}	0.4	—	$t_{AUCKMcyc}$	
$\overline{\text{AUDRST}}$ setup time (monitor mode)		$t_{AURSTMS}$	30	—	ns	
$\overline{\text{AUDRST}}$ pulse width (monitor mode)		$t_{AURSTMW}$	5	—	$t_{AUCKMcyc}$	
AUDMD setup time (monitor mode)		t_{AUMDMS}	5	—	$t_{AUCKMcyc}$	

Item	Symbol	Min.	Max.	Unit	Figures
Monitor data output delay time	t_{AUDTMD}	—	35	ns	Figure 34.37
Monitor data input setup time	t_{AUDTMS}	15	—	ns	
Monitor data input hold time	t_{AUDTMH}	5	—	ns	
$\overline{\text{AUDSYNC}}$ input setup time	t_{AUDSYS}	15	—	ns	
$\overline{\text{AUDSYNC}}$ input hold time	t_{AUDSYH}	5	—	ns	



Note: The cycle of AUDCK depends on PLL multiply ratio and the setting of AUDCK divide ratio
The phase of between CK and AUDCK is independent.

Figure 34.34 Trace Mode Reset Timing

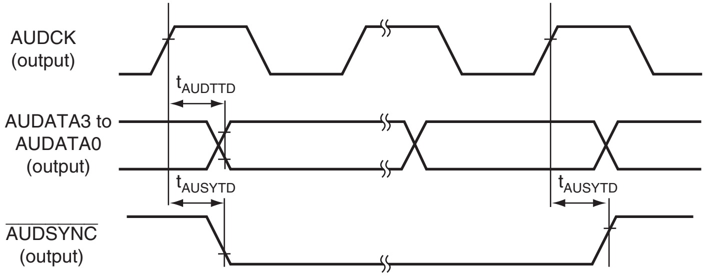


Figure 34.35 Trace Mode Timing

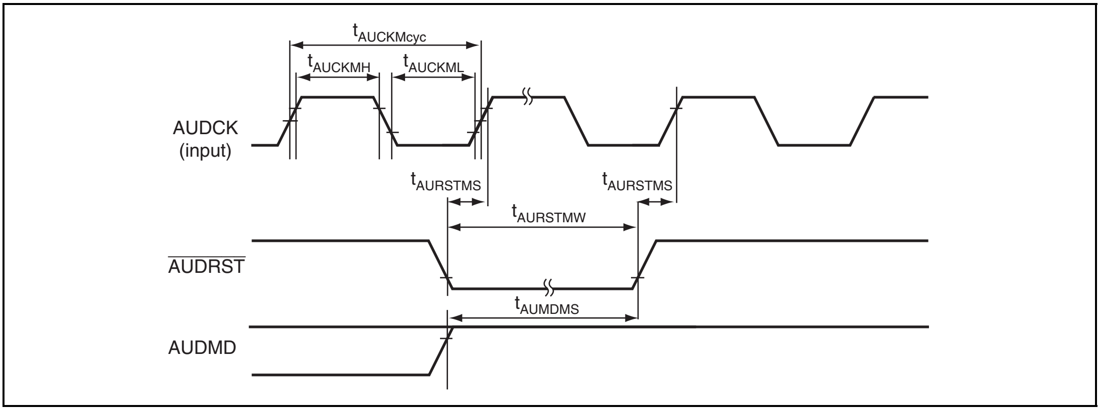


Figure 34.36 Monitor Mode Reset Timing

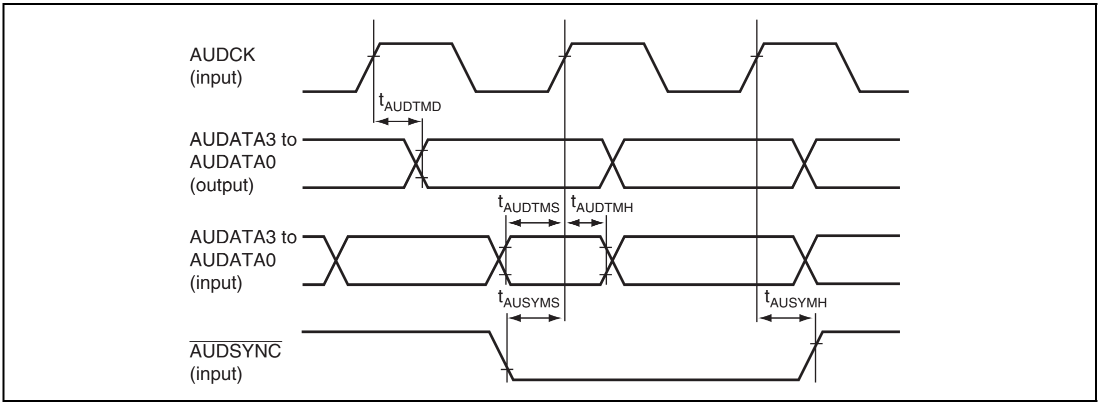
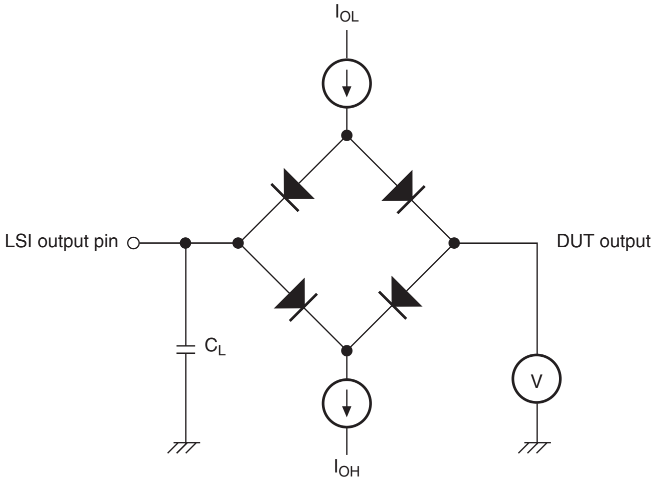


Figure 34.37 Monitor Mode Timing

34.3.18 Measuring Conditions for AC Characteristics

Input reference levels High level: V_{IH} min. value, low level: V_{IL} max. value
Output reference level High level: 2.0 V, Low level: 0.8 V
Input rise and fall times: 1ns



Notes: 1. C_L is the total value that includes the capacitance of measurement tools.
Each pin is set as follows:
30pF: CK
50pF: All other pins
2. I_{OH} = 200 µA, I_{OL} = 1.6 mA

Figure 34.38 Output Test Circuit

34.4 A/D Converter Characteristics

Table 34.30 shows the A/D converter characteristics.

Table 34.30 A/D Converter Characteristics

Conditions: V_{CC} = PLLV_{CC} = 3.3 V ±0.3 V, PV_{CC}1 = 5.0 V ±0.5 V/3.3 V ±0.3 V,
PV_{CC}2 = 5.0 V ±0.5 V,
AV_{CC} = 5.0 V ±0.5 V, AVREFH_A = AVREFH_B = 4.5 V to AV_{CC},
V_{SS} = PLLV_{SS} = AV_{SS} = AVREFL_A = AVREFL_B = 0 V
When PV_{CC}1 = 3.3 V ±0.3 V, V_{CC} = PV_{CC}1.
T_a = –40°C to 125°C

Item		Symbol	Min.	Max.	Unit	Figures
Digit resolution		—	12	12	bit	
Voltage resolution*[1]		—	1.10	1.34	mV	
A/D conversion cycle*[2]		—	25	50	t_{cyc}	
A/D conversion time (f_{OP} = 20[MHz]/40[MHz])		—	1.25	2.5	μs	
Non-linear error		—	—	±4.0	LSB	
Offset error		—	—	±7.5	LSB	
Full-scale error		—	—	±7.5	LSB	
Quantization error		—	—	±0.5	LSB	
Absolute error		—	—	±8.0	LSB	
Absolute error at self-diagnosis		—	—	±8.0	LSB	
Analog input capacitance	Awaiting	—	—	20	pF	
	Sampling	—	—	40	pF	
Permitted analog signal source impedance		—	—	3	kΩ	

Notes: 1. At AVREFH-AVREFL = 4.5 V, resolution is 1.10mV.
At AVREFH-AVREFL = 5.5 V, resolution is 1.34mV.
2. A/D conversion cycle depends on CKS bit setting in ADCER0/1 register.

34.5 Flash Memory Characteristics

Table 34.31 shows the flash memory characteristics. The characteristics are measured under the condition that the frequency of the clock signal input on the EXTAL pin is 20 MHz.

Table 34.31 Flash Memory Characteristics

Conditions: V_{CC} = PLLV_{CC} = 3.3 V ±0.3 V, PV_{CC}1 = 5.0 V ±0.5 V/3.3 V ±0.3 V,
PV_{CC}2 = 5.0 V ±0.5 V,
AV_{CC} = 5.0 V ±0.5 V, AVREFH_A = AVREFH_B = 4.5 V to AV_{CC},
V_{SS} = PLLV_{SS} = AV_{SS} = AVREFL_A = AVREFL_B = 0 V
When PV_{CC}1 = 3.3 V ±0.3 V, V_{CC} = PV_{CC}1.
T_a = –40°C to 125°C

Item			Symbol	Min.	Typ.	Max.	Unit	Figures
Programming time	256 bytes	Programming/erasing: 100 times or less	t_{P256}	—	2	12	ms	
		Programming/erasing: 101 to 1000 times		—	2.4	14.4		
	8 Kbytes	Programming/erasing: 100 times or less	t_{P8K}	—	45	100	ms	
		Programming/erasing: 101 to 1000 times		—	54	120		
Erase time	8 Kbytes	Programming/erasing: 100 times or less	t_{E8K}	—	50	120	ms	
		Programming/erasing: 101 to 1000 times		—	60	144		
	64 Kbytes	Programming/erasing: 100 times or less	t_{E64K}	—	400	875	ms	
		Programming/erasing: 101 to 1000 times		—	480	1050		
	128 Kbytes	Programming/erasing: 100 times or less	t_{E128K}	—	800	1750	ms	
		Programming/erasing: 101 to 1000 times		—	960	2100		
	32 Kbytes*	Programming/erasing: 100 times or less	t_{E32K}	—	200	480	ms	
		Programming/erasing: 101 to 1000 times		—	240	576		

Item	Symbol	Min.	Typ.	Max.	Unit	Figures
Reprogramming/Erasing cycle	N_{PEC}	—	—	1000	Times	
Suspend delay during programming	t_{SPD}	—	—	120	µs	Figure 34.39
1st time suspend delay during erasing when suspend priority mode	t_{SESD1}	—	—	120	µs	
2nd time suspend delay during erasing when suspend priority mode	t_{SESD2}	—	—	1.7	ms	
Suspend delay during erasing when erasing priority mode	t_{SEED}	—	—	1.7	ms	

Note: * User Boot Mat is 32 Kbytes.

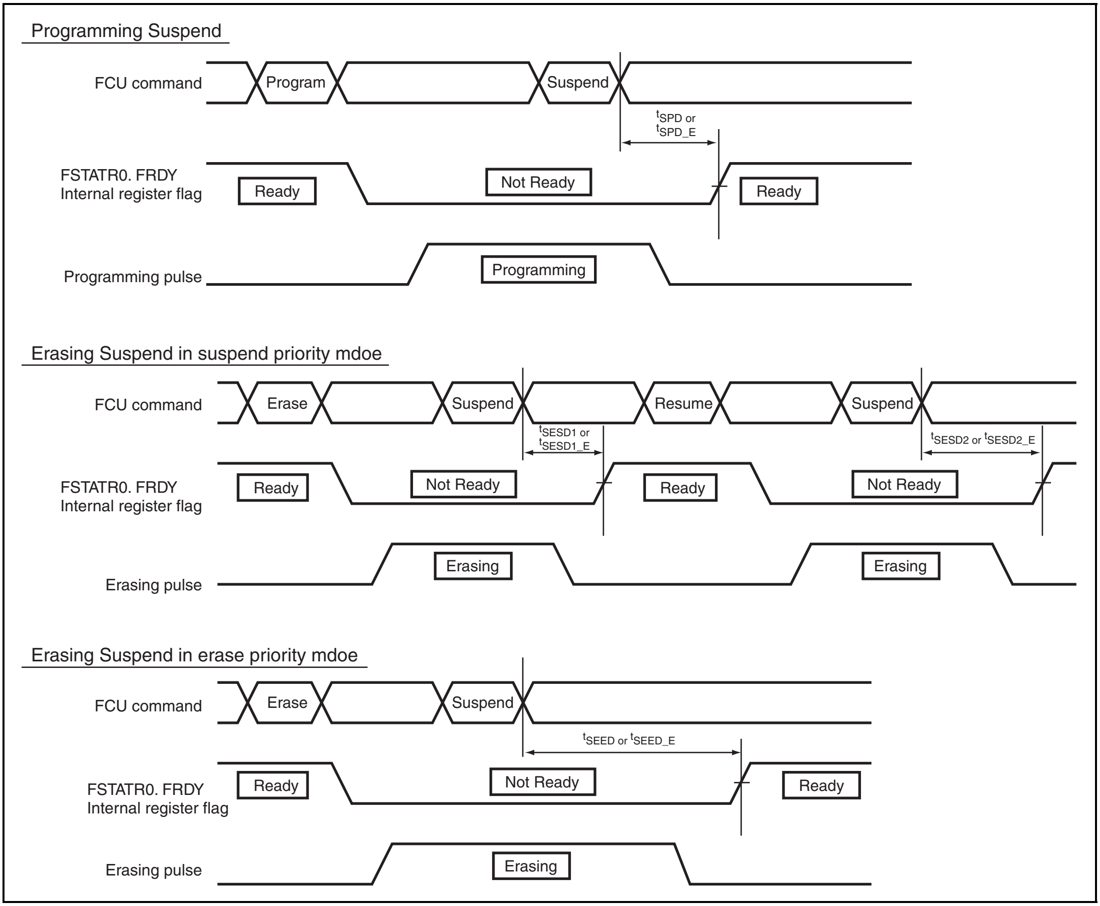


Figure 34.39 Flash Programming/Erasing Suspend Timing

34.6 EEPROM Characteristics

Table 34.32 shows the EEPROM characteristics. The characteristics are measured under the condition that the frequency of the clock signal input on the EXTAL pin is 20 MHz.

Table 34.32 EEPROM Characteristics

Conditions: V_{CC} = PLLV_{CC} = 3.3 V ±0.3 V, PV_{CC}1 = 5.0 V ±0.5 V/3.3 V ±0.3 V,
PV_{CC}2 = 5.0 V ±0.5 V,
AV_{CC} = 5.0 V ±0.5 V, AVREFH_A = AVREFH_B = 4.5 V to AV_{CC},
V_{SS} = PLLV_{SS} = AV_{SS} = AVREFL_A = AVREFL_B = 0 V
When PV_{CC}1 = 3.3 V ±0.3 V, V_{CC} = PV_{CC}1.
T_a = –40°C to 125°C

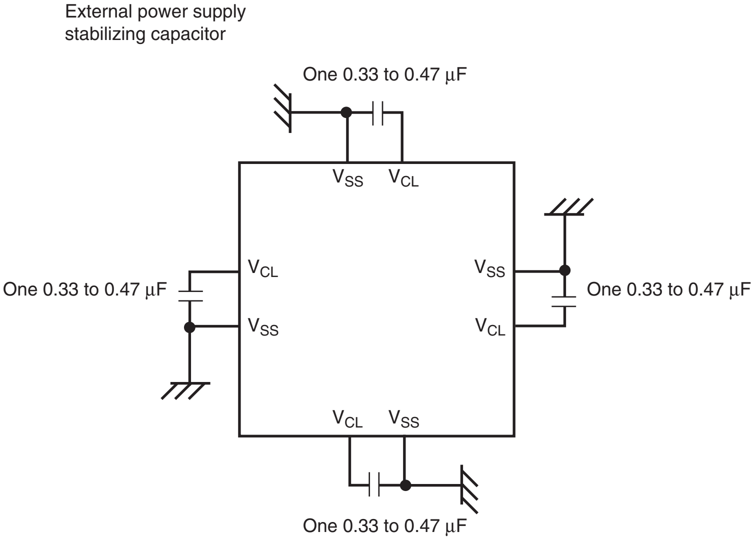
Item			Symbol	Min.	Typ.	Max.	Unit	Figures
Programming time	8 bytes		t_{P8_E}	—	0.4	2	ms	
	128 bytes		t_{P128_E}	—	1.0	5.0	ms	
Erase time	8 Kbytes	Programming/erasing: 100 times or less	t_{E8K_E}	—	80	250	ms	
		Programming/erasing: 101 to 50000 times		—	250	750		
Blank check time	8 bytes		t_{BC8_E}	—	—	30	µs	
	8 Kbytes		t_{BC8K_E}	—	—	2.5	ms	
Reprogramming/erasing cycle			N_{PEC_E}	—	—	50000*	Times	
Suspend delay during programming			t_{SPD_E}	—	—	120	µs	Figure 34.39
1st time suspend delay during erasing when suspend priority mode			t_{SESD1_E}	—	—	120	µs	
2nd time suspend delay during erasing when suspend priority mode			t_{SESD2_E}	—	—	1.7	ms	
Suspend delay during erasing when erasing priority mode			t_{SEED_E}	—	—	1.7	ms	

Note: * In case that the programming/erasing cycle is 30000 times or less, the data retention period is 15 years.
In case that the programming/erasing cycle is 30001 to 50000 times, the data retention period is 2 years.

34.7 Usage Note

34.7.1 Notes on Connecting External Capacitor for Current Stabilization

This LSI includes an internal step-down circuit to automatically reduce the microprocessor power supply voltage to an appropriate level. Between this internal stepped-down power supply (V_{CL} pin) and the V_{SS} pin, a capacitor for stabilizing the internal voltage. Connection of the external capacitor is shown in figure 34.40. The external capacitor should be located near the pin. Do not apply any power supply voltage to the V_{CL} pin.



Note: Do not apply any power supply voltage to the V_{CL} pin.
Use multilayer ceramics capacitors (one 0.33 to 0.47 µF capacitor for each V_{CL} pin), which should be located near the pin.

Figure 34.40 Connection of V_{CL} Capacitor

Table 34.33 Capacitance for Internal Voltage Stabilization

Item		Symbol	Min.	Max.	Unit
Capacitance for internal voltage stabilization (V_{CL} pins)	Per pin	CV_{CL}	0.33	0.47	μF
	Total	ΣCV_{CL}	1.32	1.88	μF

Appendix

A. Pin States

Tables A.1 to A.3 show the pin states of SH7256.

Table A.1 Pin States

Pin Function		Pin State				
		Power-On Reset State				Power-Down Mode
		ROM Disabled Extension Mode				
Classification	**Abbreviation**	**8 Bits**	**16 Bits**	**ROM Enabled Extension Mode**	**Single-Chip Mode**	**Hardware Standby**
Clock	CK	O	O	O	O	Z
	XTAL	O	O	O	O	L
	EXTAL	I	I	I	I	I
System control	$\overline{\text{HSTBY}}$	I	I	I	I	I
	$\overline{\text{RES}}$	I	I	I	I	Z
	ASEMD	I (pulled down)	I (pulled down)	I (pulled down)	I (pulled down)	I
	MD_CLKP	I	I	I	I	I
	MD_CLK1, MD_CLK0	I	I	I	I	I
	MD4	I	I	I	I	I
	MD3	I	I	I	I	I
	FWE	I	I	I	I	I
	MD2 to MD0	I	I	I	I	I
	$\overline{\text{WDTOVF}}$	O	O	O	O	Z
Interrupt	NMI	I	I	I	I	Z
	$\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$	—	—	—	—	—
Address bus	A21 to A0	O	O	O	—	Z
Data bus	D7 to D0	I (pulled up)	I (pulled up)	I (pulled up)	—	Z
	D15 to D8	—	I (pulled-up)	I (pulled-up)	—	Z

Pin Function		Pin State				
		Power-On Reset State				Power-Down Mode
		ROM Disabled Extension Mode				
Classification	Abbreviation	8 Bits	16 Bits	ROM Enabled Extension Mode	Single-Chip Mode	Hardware Standby
Bus control	$\overline{\text{WAIT}}$	I	I	I	—	Z
	$\overline{\text{WE1}}$	—	H	H	—	Z
	$\overline{\text{WE0}}$	H	H	H	—	Z
	$\overline{\text{RD}}$	H	H	H	—	Z
	RD/$\overline{\text{WR}}$	H	H	H	—	Z
	$\overline{\text{CS3}}$ to $\overline{\text{CS1}}$	—	—	—	—	—
	$\overline{\text{CS0}}$	H	H	H	—	Z
Ports	$\overline{\text{POD}}$	—	—	—	—	—
ATU-III	TIA05 to TIA00	—	—	—	—	—
	TIOC43 to TIOC40, TIOC33 to TIOC30, TIOC23 to TIOC20, TIOC13 to TIOC10, TIOC03 to TIOC00	—	—	—	—	—
	TOD33A to TOD30A, TOD23A to TOD20A, TOD13A to TOD10A, TOD03A to TOD00A	—	—	—	—	—

Pin Function		Pin State				
		Power-On Reset State				Power-Down Mode
		ROM Disabled Extension Mode				
Classification	Abbreviation	8 Bits	16 Bits	ROM Enabled Extension Mode	Single-Chip Mode	Hardware Standby
ATU-III	TOD33B to TOD30B, TOD23B to TOD20B, TOD13B to TOD10B, TOD03B to TOD00B	—	—	—	—	—
	TOE63 to TOE60, TOE53 to TOE50, TOE43 to TOE40, TOE33 to TOE30, TOE23 to TOE20, TOE13 to TOE10, TOE03 to TOE00	—	—	—	—	—
	TIF2A to TIF0A	—	—	—	—	—
	TIF2B to TIF0B	—	—	—	—	—
	TIF25 to TIF3	—	—	—	—	—
	TIJ1, TIJ0	—	—	—	—	—
	TCLKA, TCLKB	—	—	—	—	—
SCI	SCK_A to SCK_E	—	—	—	—	—
	TxD_A to TxD_E	—	—	—	—	—

Pin Function		Pin State				
		Power-On Reset State				Power-Down Mode
		ROM Disabled Extension Mode				
Classification	Abbreviation	8 Bits	16 Bits	ROM Enabled Extension Mode	Single-Chip Mode	Hardware Standby
SCI	RxD_A to RxD_E	—	—	—	—	—
RSPI	RSPCKA to RSPCKC	—	—	—	—	—
	MOSIA to MOSIC	—	—	—	—	—
	MISOA to MISOC	—	—	—	—	—
	SSLA0 to SSLC0	—	—	—	—	—
	SSLA1 to SSLA7	—	—	—	—	—
	SSLB1 to SSLB3	—	—	—	—	—
	SSLC1 to SSLC3	—	—	—	—	—
A/D converter	AN_A27 to AN_A0	Z	Z	Z	Z	Z
	AN_B48 to AN_B40	Z	Z	Z	Z	Z
	ADTRG_A, ADTRG_B	—	—	—	—	—
	ADEND_A, ADEND_B	—	—	—	—	—
	AVREFH_A, AVREFH_B	I	I	I	I	I
	AVREFL_A, AVREFL_B	I	I	I	I	I

Pin Function		Pin State				
		Power-On Reset State				Power-Down Mode
		ROM Disabled Extension Mode				
Classification	Abbreviation	8 Bits	16 Bits	ROM Enabled Extension Mode	Single-Chip Mode	Hardware Standby
RCAN	CTx_A to CTx_D	—	—	—	—	—
	CRx_A to CRx_D	—	—	—	—	—
FlexRay	FRTxD_A, AFRTxD_B	—	—	—	—	—
	FREN_A, AFREN_B	—	—	—	—	—
	FRRxD_A, AFRRxD_B	—	—	—	—	—
UBC	$\overline{\text{UBCTRG}}$	—	—	—	—	—
I/O ports	PA15 to PA0	—	—	—	I	Z
	PB14	—	—	—	I	Z
	PB13 to PB11	I	I	I	I	Z
	PB10 to PB8	—	—	—	I	Z
	PB7	I	—	—	I	Z
	PB6	—	—	—	I	Z
	PB5 to PB0	—	—	—	I	Z
	PC15 to PC8	I	—	—	I	Z
	PC7 to PC0	—	—	—	I	Z
	PD13 to PD0	I	I	I	I	Z
	PE13 to PE0	I	I	I	I	Z
	PF15 to PF0	I	I	I	I	Z
	PG15 to PG0	I	I	I	I	Z
	PH5 to PH0	I	I	I	I	Z
	PJ9 to PJ0	I	I	I	I	Z
	PK11 to PK0	I	I	I	I	Z
	PL8 to PL0	I	I	I	I	Z

Table A.2 Pin States for JTAG Interface

Pin Function		Pin State						
		Power-On Reset State				Power-Down Mode		
		ROM Disabled Extension Mode						
Classification	Abbreviation	8 Bits	16 Bits	ROM Disabled Extension Mode	Single-Chip Mode	Hardware Standby	Module Standby	Left Unconnected
JTAG	TMS	I	I	I	I	Z	Z	Pulled up
	$\overline{\text{TRST}}$	I	I	I	I	Z	Z	Pulled up
	TDI	I	I	I	I	Z	Z	Pulled up
	TDO	O/Z	O/Z	O/Z	O/Z	Z	Z	O/Z
	TCK	I	I	I	I	Z	Z	Pulled up

Table A.3 Pin States for Advanced User Debugger (AUD)

Pin Function		Pin State						
		Power-Down State		AUD Reset ($\overline{\text{AUDRST}}$ = L)		AUD in Operation		
Classification	Abbreviation	Hardware Standby	AUD Module Standby	AUDMD = H	AUDMD = L	AUDMD = H	AUDMD = L	Left Unconnect-ed
AUD	$\overline{\text{AUDRST}}$	Z	Z	L input (pulled down)	L input (pulled down)	H input (pulled down)	H input (pulled down)	Pulled down
	AUDMD	Z	Z	I (pulled up)	I (pulled up)	I (pulled up)	I (pulled up)	Pulled up
	AUDATA3 to AUDATA0	Z	Z	I (pulled up)	pulled up	I/O (pulled up)	O	Pulled up
	AUDCK	Z	Z	I (pulled up)	pulled up	I (pulled up)	O	Pulled up
	$\overline{\text{AUDSYNC}}$	Z	Z	I (pulled up)	pulled up	I (pulled up)	O	Pulled up

[Legend]

—: Initial value undefined

I: Input

O: Output

H: High level output

L: Low level output

Z: High impedance

Pulled up: Pulled up by the resistors inside the LSI

Pulled down: Pulled down by the resistors inside the LSI

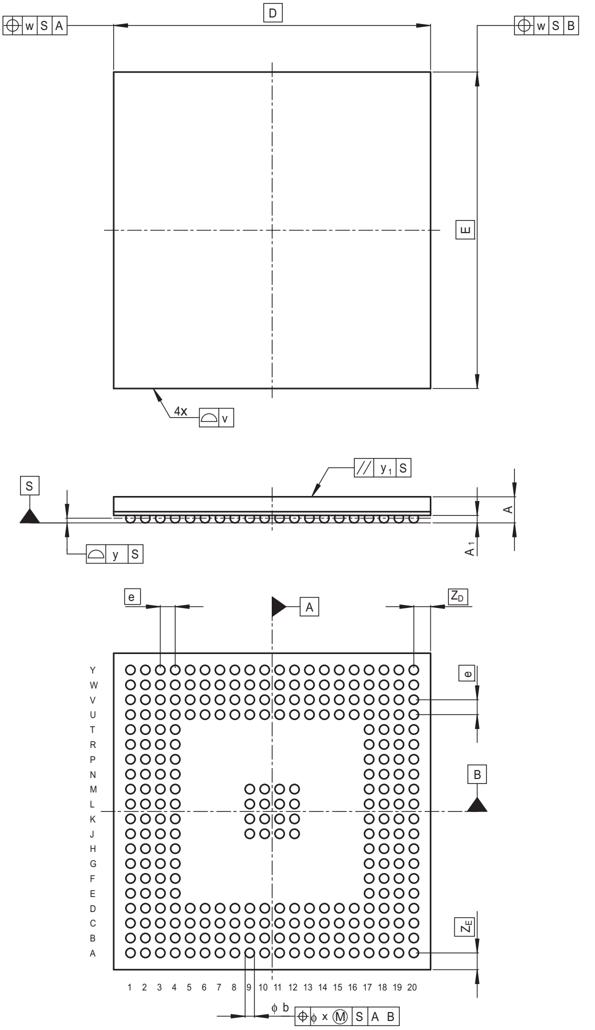
B. Product Code Lineup

Table B.1 Product Code Lineup

Product Type	Product Code	Package
SH72567R	R5F72567RKBGV	PRBG0272FA-A (BP-272)

C. Package Dimensions

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-BGA272-21x21-1.00	PRBG0272FA-A	BP-272/BP-272V	1.3g



Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	——	21.0	——
E	——	21.0	——
v	——	——	0.20
w	——	——	0.30
A	——	——	2.10
A_1	0.36	0.46	0.56
e	——	1.00	——
b	0.53	0.63	0.73
x	——	——	0.10
y	——	——	0.15
y_1	——	——	0.35
S_D	——	——	——
S_E	——	——	——
Z_D	——	1.00	——
Z_E	——	1.00	——

Figure C.1 Package Dimensions

Main Revisions and Additions in this Edition

Item	Page	Revision (See Manual for Details)
—	—	Pin name amended [Before amendment] RD$\overline{\text{WR}}$ → [After amendment] RD/$\overline{\text{WR}}$
Figure 1.2 Pin Arrangements	12	Description added Top View
Table 3.2 Operating Mode Pin Settings	86	Pin name amended [Before amendment] FEW → [After amendment] FWE
Table 4.5 Crystal Resonator Parameters (Recommended Values)	91	Parameters amended Rs ~~Max.~~ (Ω) Co ~~Max.~~ (pF)
6.1 Reset Operation	99	Amended When the $\overline{\text{RES}}$ pin is driven low by the low level pulse longer than or equal to the noise cancellation width (t_{RESNCW}), the reset request is accepted. When the reset request is accepted, all pin states are reset and each pin enters reset state. Each pin state during reset is summarized in appendix A, Pin States.
9.5 Usage Notes	234	Deleted 9. Do not place a pre-execution break for the instruction immediately after a DIVU (or DIVS) instruction. Otherwise the break will be still taken even though an exception or interrupt occurs during the DIVU (or DIVS) instruction execution. ~~10. Do not place a post-execution break for a DIVU (or DIVS) instruction. Otherwise the break will be still taken even though an exception or interrupt occurs during the DIVU (or DIVS) instruction execution.~~
Table 10.7 8-Bit External Device Access and Data Alignment	253	Table title amended

Item	Page	Revision (See Manual for Details)
12.3.2 Each A-DMAC Channel Operation (4) Operation for A-DMAC Channels Used for RCAN • Operation Details and Transfer flow	392	Amended 5. Sets the TE bit to 1 if all the TV bits are cleared to 0 and if the specified transfer has been completed. In this case, if an interrupt is enabled (IE = 1), an interrupt is requested to the CPU. 6. While the TE bit is set to 1, transfer is not performed even if the TV bit is set.
13.12 Overview of Timer B (2) Frequency-Multiplied Clock Generator	490	Amended The frequency-multiplied clock generator generates a clock signal by producing from 1 to 4095 cycles in response to an external-event input signal.
13.16.3 Timer Control Registers C0 to C4 (TCRC0 to TCRC4)	535	Bit 3, Description amended In PWM mode, do not set GRCn0 to GRCn3 to H'000000. If GRCn0 is set to H'000000, compare match occurs at illegal cycles.
13.27.2 Interrupt Requests	662	Amended and added Six timer G interrupts, CMIG0 to CMIG5, are available in timer G. When a compare match is detected in the subblocks, an interrupt request is output. The interrupt request is received in the direct memory access controller (DMAC) and interrupt controller (INTC).
Figure 13.46 Operation Example of Counter and Compare Match	669	Figure replaced
13.34.3 FIFO Control Registers J0 and J1 (FCRJ0 and FCRJ1)	684	Note 2 added
Section 16 Serial Communications Interface (SCI)	—	Term amended [Before amendment] SCIF → [After amendment] SCI
Figure 16.11 Example of SCI Transmit Operation	806	Figure replaced

Item	Page	Revision (See Manual for Details)
18.2 Architecture	887	Amended The RCAN-TL1 device offers a flexible and sophisticated way to organise and control CAN frames, providing the compliance to CAN2.0B Active and ISO-11898-1. The module is formed from 7 different functional entities. These are the Micro Processor Interface (MPI), Mailbox, Mailbox Control, Timer, CAN Interface, Parity Control, and Parity Circuit. The figure below shows the block diagram of the RCAN-TL1 Module. The bus interface timing is designed according to the peripheral bus I/F required for each product.
18.7 DMAC Interface	990	Amended The DMAC can be activated by the reception of a message in RCAN-TL1 mailbox 0. When DMAC transfer ends after DMAC activation has been set, flags of RXPR0 and RFPR0 are cleared automatically. An interrupt request due to a receive interrupt from the RCAN-TL1 cannot be sent to the CPU in this case. Figure 18.27 shows a DMAC transfer flowchart.
18.8 CAN Bus Interface	991	Amended A bus transceiver IC is necessary to connect this LSI to a CAN bus. A Renesas HA13721 transceiver IC and its compatible products are recommended. Figure 18.28 shows a sample connection diagram.
21.3.3 Bypass Register (SDBPR)	1309	Deleted SDBPR is a 1-bit register and is connected between the TDI and TDO pins when SDIR is set to BYPASS mode. ~~The initial value is undefined.~~ The value in SDBPR is not initialized by a power-on reset or $\overline{\mathrm{TRST}}$ pin assertion, but the first value read by the BYPASS instruction is always 0 because this register is initialized in the Capture-DR state of this instruction.
23.1.5 Port B Control Registers 1 to 4 (PBCR1 to PBCR4) (2) Port B Control Register 3 (PBCR3)	1406	Bits 13, 12, Description amended • Extension mode 00: PB11 input/output (port) (initial value) 01: Setting prohibited 10: $\overline{\mathrm{CS1}}$ output (BSC) 11: TOE21 ~~input/~~output (ATU-III) • Single-chip mode 00: PB11 input/output (port) (initial value) 01: Setting prohibited 10: PB11 input/output (port) 11: TOE21 ~~input/~~output (ATU-III)

Item	Page	Revision (See Manual for Details)
24.11.3 Port J Inverting Register (PJIR)	1558	Amended Setting the specific bits of PJIR to 1 inverts the output value on their corresponding pins.
26.5.2 State Transition in Boot Mode (4) Waiting for Host Command for Programming or Erasure	1628, 1629	Note added
26.5.4 Inquiry/Selection Host Command Wait State (1) Supported Device Inquiry	1633	Amended Product code (n bytes): ASCII code for the name of the supported device
(9) Erasure Block Information Inquiry	1641	Appearance of Response amended
26.5.5 Programming/Erasing Host Command Wait State (9) User Boot MAT Blank Check (10) User MAT Blank Check	1657	Added No verification function is provided to check program/erase state of the area where the data is undefined by suspend of program/erase (e.g., reset input, power-supply interruption). Therefore, if the undefined area should be used again, make sure to completely erase data before usage.
(11) Read Lock Bit Status (12) Lock Bit Program	1658, 1659	Replaced
26.6.3 FCU Command Usage	1671	Added … may have occurred. In such case, initialize the FCU by an FCU reset. If the FRDY is set to 1 upon completion of the FCU command handling, while the FRDCLE and FRCCLE bits are 1, the FCUERR bit and the FRDTCT and FRCRCT bits are also 0. Therefore, …
28.1 Features • Blank check function	1730	Added Blank check function checks the erase state of the area where erase has ended. The function is disabled when programming/erasing was suspended (e.g., reset input, power-supply interruption).

<table>
<tr><th>Item</th><th>Page</th><th>Revision (See Manual for Details)</th></tr>
<tr><td>28.9 Usage Notes
(5) Reset during Programming or Erasure</td><td>1771</td><td>Added
No verification function is provided to check program/erase state of the area where the data is undefined by suspend of program/erase (e.g., reset input, power-supply interruption). Therefore, if the undefined area should be used again, make sure to completely erase data before usage.</td></tr>
<tr><td>33.1 Register Addresses (grouped by module name, in ordered of the corresponding section numbers)</td><td>1845 to 1847</td><td>TEMSK0 to TEMSK7, Register Name amended
DMA TE flag mask setting register 0
DMA TE flag mask setting register 1
DMA TE flag mask setting register 2
DMA TE flag mask setting register 3
DMA TE flag mask setting register 4
DMA TE flag mask setting register 5
DMA TE flag mask setting register 6
DMA TE flag mask setting register 7</td></tr>
<tr><td></td><td>1848</td><td>ADMAABR, Register Name amended
A-DMAC alias base register</td></tr>
<tr><td>Table 34.2 Correspondence between Power Supply Names and Pins</td><td>2208</td><td>Deleted
<table>
<tr><th>Pin No.</th><th>Symbol</th><th>Permissible Input Voltage (V)</th></tr>
<tr><td>B11</td><td>$\overline{\text{WDTOVF}}$</td><td>~~V_{cc} + 0.3~~</td></tr>
</table></td></tr>
<tr><td>Table 34.5 DC Characteristics Input Leak Current</td><td>2226</td><td>Amended
<table>
<tr><th colspan="2">Item</th><th>Measurement Conditions</th></tr>
<tr><td>Input leak current</td><td>$\overline{\text{HSTBY}}$, $\overline{\text{RES}}$, NMI, ASEMD, FWE, MD4 to MD0, MD_CLK1, MD_CLK0, MD_CLKP, EXTAL</td><td>Vin = 0.3 V to V_{CC} − 0.3 V</td></tr>
<tr><td></td><td>PA15 to PA0, PB14 to PB0 (extended bus mode)</td><td>Vin = 0.3 V to $PV_{CC}1$ − 0.3 V,
$PV_{CC}1$ = V_{CC} = 3.3 V ± 0.3 V</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Item</th><th>Page</th><th>Revision (See Manual for Details)</th></tr>
<tr><td>Table 34.5 DC Characteristics Input Leak Current</td><td>2227</td><td>Amended
<table>
<tr><th colspan="2">Item</th><th>Measurement Conditions</th></tr>
<tr><td>Input leak current</td><td>A/D port</td><td>Vin = 0.3 V to AV_{CC} – 0.3 V,
T_a = –40°C to 105°C</td></tr>
<tr><td></td><td></td><td>Vin = 0.3 V to AV_{CC} – 0.3 V,
T_a = 105°C to 125°C</td></tr>
</table></td></tr>
<tr><td>Table A.1 Pin States</td><td>2279</td><td>Deleted
<table>
<tr><th>Classification</th><th>Abbreviation</th><th>Hardware Standby</th></tr>
<tr><td>System control</td><td>ASEMD</td><td>I ~~(pulled down)~~</td></tr>
</table></td></tr>
</table>

Index

SH7256 Group User’s Manual: Hardware

Publication Date: Rev.1.00 Dec 24, 2010
Rev.3.00 Apr 01, 2014

Published by: Renesas Electronics Corporation

SALES OFFICES

Renesas Electronics Corporation

http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100191, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, P. R. China 200333
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-Ku, Seoul, 135-920, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 3.0

SH7256 Group

Renesas Electronics Corporation